Radboud Repository

Radboud University Nijmegen

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/199893

Please be advised that this information was generated on 2019-06-02 and may be subject to change.

Proof terms for generalized natural deduction

Herman Geuvers
Radboud University Nijmegen & Technical University Eindhoven, The Netherlands
herman@cs.ru.nl

Tonny Hurkens
Author: Please enter affiliation as second parameter of the author macro
hurkens@science.ru.nl

Abstract

In previous work it has been shown how to generate natural deduction rules for propositional connectives from truth tables, both for classical and constructive logic. The present paper extends this for the constructive case with proof-terms, thereby extending the Curry-Howard isomorphism to these new connectives. A general notion of conversion of proofs is defined, both as a conversion of derivations and as a reduction of proof-terms. It is shown how the well-known rules for natural deduction (Gentzen, Prawitz) and general elimination rules (Schroeder-Heister, von Plato, and others), and their proof conversions can be found as instances. As an illustration of the power of the method, we give constructive rules for the nand logical operator (also called *Sheffer stroke*).

As usual, conversions come in two flavours: either a *detour conversion* arising from a *detour convertibility*, where an introduction rule is immediately followed by an elimination rule, or a *permutation conversion* arising from an *permutation convertibility*, an elimination rule nested inside another elimination rule. In this paper, both are defined for the general setting, as conversions of derivations and as reductions of proof-terms. The properties of these are studied as proof-term reductions. As one of the main contributions it is proved that detour conversion is strongly normalizing and permutation conversion is strongly normalizing: no matter how one reduces, the process eventually terminates. Furthermore, the combination of the two conversions is shown to be weakly normalizing: one can always reduce away all convertibilities.

2012 ACM Subject Classification Theory of computation → Proof theoryTheory of computation → Type theoryTheory of computation → Constructive mathematicsTheory of computation → Functional constructs

Keywords and phrases constructive logic, natural deduction, detour conversion, permutation conversion, normalization Curry-Howard isomorphism

Digital Object Identifier 10.4230/LIPIcs.TYPES.2017.4

Acknowledgements We thank Iris van der Giessen and the anonymous referees for their valuable comments on the earlier version of this paper.

1 Introduction

Natural deduction rules come in various forms, where the tree format is the most well-known. One either puts formulas A as the nodes and leaves of the tree, or sequents $\Gamma \vdash A$, where Γ is a sequence or a finite set of formulas. Other formalisms use a linear format, using flags or boxes to explicitly manage the open and discharged assumptions.

We [7] use a natural deduction in sequent calculus style, where in addition all rules have a special form:

$$\frac{\ldots \quad \Gamma \vdash A_i \quad \ldots \qquad \ldots \quad \Gamma, A_j \vdash D \quad \ldots}{\Gamma \vdash D}$$

© Herman Geuvers and Tonny Hurkens;
licensed under Creative Commons License CC-BY

So if the conclusion of a rule is $\Gamma \vdash D$, then the hypotheses of the rule can be of one of two forms:

1. $\Gamma, A_j \vdash D$: we still need to prove D from Γ, but we are now also allowed to use A_j as additional assumption. We call A_j a case.
2. $\Gamma \vdash A_i$: in stead of proving D from Γ, we now need to prove A_i from Γ. We call A_i a lemma.

Given the restricted format of the rules, we don't have to give Γ explicitly, as it can be retrieved from the other information in a deduction. So, the deduction rules are presented without Γ, in the following format

$$\frac{\dots \quad \vdash A_i \quad \dots \qquad \dots \quad A_j \vdash D \quad \dots}{\vdash D}$$

In [7] we have shown how to derive natural deduction rules for a connective form its definition by a truth table, both for the classical and the intuitionistic case. In that paper, we have shown that the intuitionistic rules are indeed constructive by providing a Kripke semantics. In the present paper we provide a proof-theoretic study of the natural deduction rules for the intuitionistic case. We define a notion of convertibility and conversion for the general connectives, which we analyze by interpreting derivations as proof-terms. So we extend the Curry-Howard isomorphism, that interprets formulas as types and derivations as terms, to include all these new intuitionistic connectives.

It turns out that the standard format for the deduction rules we have chosen (as described above) is very suitable for defining convertibilities and conversion in general, for giving a term interpretation to derivations and for defining reductions on these proof-terms that correspond with conversion (both detour conversion and permutation conversion). The format of our rules also allows the transformation of other formalisms, like the very well-known ones by Gentzen and Prawitz [6, 14] but also more recent ones by Von Plato [23], in terms of ours. This transformation we will define on the proof-term level and we will show how *detour conversion* (the elimination of a *direct convertibility*, an introduction rule immediately followed by an elimination rule) is preserved by the translation.

Standard questions about logic are consistency and decidability. We prove that both hold (in general for our connectives) by proving *weak normalization* for the combined process of *detour conversion* and *permutation conversion*. A permutation conversion operates on a *permutation convertibility*, which arises when an elimination rule blocks a detour convertibility for another connective; in that case one has to permute one elimination rule over another. Weak normalization states that for any derivation (proof-term) we can eliminate convertibilities in such a way that eventually no convertibilities are left. Using this one can prove the sub-formula property and consistency and decidability. We prove weak normalization for the proof-terms by studying reduction of proof-terms.

The interest of our work lies in the fact that the natural deduction rules can be defined and analyzed in such a generic way, capturing very many known instances of deduction rules for intuitionistic logic, but also new deduction rules for new connectives. The key concepts that make this work are our general rule format (described above) and the fact that our system provides natural deduction rules for each connective *in isolation*: rules for one connective do not use another connective. We will illustrate this by giving the nand operator as an extended example. We describe its constructive derivation rules, as they arise from the truth tables. These rules are self-contained, so they only refer to nand itself, and we show how to interpret intuitionistic proposition logic in the logic with only nand. We also give the proof-terms and reductions for nand.

1.1 Related work and contribution of the paper

Natural deduction has been studied extensively, since the original work by Gentzen [6], both for classical and intuitionistic logic. Overviews can be found in [22] and [12]. Also the generalization of natural deduction to include other connectives or allow different derivation rules has been studied by various researchers. Notably, there is the work of Schroeder-Heister [17], Read [16], Tennant [21], Von Plato [23, 12], Milne [11], Francez and Dyckhoff [4, 3] that is related to ours. Schroeder-Heister studies general formats of natural deduction where also rules may be discharged (as opposed to the normal situation where only formulas may be discharged). He also studies a general rule format for intuitionistic logic and shows that the connectives $\wedge, \vee, \rightarrow, \perp$ are complete for it. Von Plato, Milne, Francez and Dyckhoff, Read and Tennant study "general elimination rules", where the idea is that elimination rules arise naturally from the introduction rules, following Prawitz's [15] inversion principle: "the conclusion obtained by an elimination does not state anything more than what must have already been obtained if the major premise of the elimination was inferred by an introduction". The elimination rules obtained have the same flavor as the elimination rules we derive from truth tables: the conclusion of elimination Φ is not a sub-formula of Φ, but a general formula D, where there are additional hypothesis that connect Φ and D. For the standard intuitionistic connectives the general elimination rules are quite close to ours, but $\wedge$-elimination is slightly different. Von Plato [23], Lopez-Escobar [10] and Tennant [21] study the standard intuitionistic connectives with general rules.

A difference is that we focus not so much on the rules but on the fact that we can define different and new connectives constructively. In our work, we do not take the introduction rules as primary, with the elimination rules defined from them, but we derive elimination and introduction rules directly from the truth table. Then we optimize them, which can be done in various ways, where we adhere to a fixed format for the rules. Many of the general elimination rules, for example for $\wedge$, appear naturally as a consequence of our approach of deriving the rules from the truth table.

The idea of deriving deduction rules from the truth table also occurs in the work of Milne [11], but in a slightly different way: from the introduction rules, a truth table is derived and then the classical elimination rules are derived from the truth table. For the if-then-else connective, this amounts to classical rules equivalent to ours in [7], but not optimized. We start from the truth table and derive rules for intuitionistic logic.

As remarked, the main contribution of this paper is a proof-theoretic analysis of our system of generalized natural deduction via the Curry-Howard isomorphism that interprets derivations as proof terms and conversions as reductions. We show that many known conversions and reductions are captured by our approach and we prove general normalization results. These is a lot of related work on the Curry-Howard isomorphism that our work rests on, for which we refer to [18, 8].

The present paper builds on research reported in [7]. To make this paper self-contained, we include the definitions and some basic results and examples from [7]: Section 2 repeats the main definitions of [7] in slightly expanded form, where Section 2.1 adds the new example of the nand-connective (Sheffer stroke), which is worked out in detail, especially the connection between nand-logic and intuitionistic proposition logic. Section 3 defines detour conversion and permutation conversion on derivations; the second is new. Section 4 defines the Curry-Howard isomorphism for our general natural deduction format and gives (general) proof terms for natural deductions and reduction rules on them. Section 5 shows how the general rules relate to so called "optimized" rules, which are the ones that are known from the literature for natural deduction and for proof-terms. Section 6 proves normalization results for the calculi of proof-terms. Sections 4, 5, 6 are all new; Section 2.1 is largely new and Section 3 is partially new.

2 Deriving constructive natural deduction rules from truth tables

To make this paper self contained and to fix notions and notations, we recap the main definitions from [7] and explain in detail how the elimination and introduction rules for a connective are derived from its truth table. The elimination rules have the following form. Φ is the formula we eliminate. We have $\Phi = c(A_1, \ldots, A_n)$ where c is a connective of arity n and $n = k + \ell$. The formula D is arbitrary.

$$\frac{\vdash \Phi \quad \vdash A_{i_1} \quad \ldots \quad \vdash A_{i_k} \qquad A_{j_1} \vdash D \quad \ldots \quad A_{j_\ell} \vdash D}{\vdash D}\,\text{el}$$

So, $A_{i_1}, \ldots, A_{i_k}, A_{j_1}, \ldots, A_{j_\ell}$ are the direct subformulas of $\Phi = c(A_1, \ldots, A_n)$, where some appear as "lemma" and others as "case" in the derivation rule. The (intuitionistic) introduction rules have the following form. Again, c is a connective of arity n, $\Phi = c(A_1, \ldots, A_n)$ and $n = k + \ell$. (Of course, every rule has its own specific sequence $i_1, \ldots, i_k, j_1, \ldots j_\ell$.)

$$\frac{\vdash A_{i_1} \quad \ldots \quad \vdash A_{i_k} \qquad A_{j_1} \vdash \Phi \quad \ldots \quad A_{j_\ell} \vdash \Phi}{\vdash \Phi}\,\text{in}$$

For a concrete connective c, we derive the elimination and introduction rules from the truth table, as described in the following Definition, taken from [7].

▶ **Definition 1.** Given an n-ary connective c with a truth table t_c (with 2^n rows). We write $\varphi = c(p_1, \ldots, p_n)$, where $p_1, \ldots, p_n$ are proposition letters and we write $\Phi = c(A_1, \ldots, A_n)$, where $A_1, \ldots, A_n$ are arbitrary propositions. Each row of t_c gives rise to an elimination rule or an introduction rule for c in the following way.

$$\begin{array}{ccc|c} p_1 & \ldots & p_n & c(p_1, \ldots, p_n) \\ \hline a_1 & \ldots & a_n & 0 \end{array} \quad \mapsto \quad \frac{\vdash \Phi \quad \ldots \vdash A_j (\text{if } a_j = 1) \ldots \quad \ldots A_i \vdash D (\text{if } a_i = 0) \ldots}{\vdash D}\,\text{el}$$

$$\begin{array}{ccc|c} p_1 & \ldots & p_n & c(p_1, \ldots, p_n) \\ \hline b_1 & \ldots & b_n & 1 \end{array} \quad \mapsto \quad \frac{\ldots \vdash A_j (\text{if } b_j = 1) \ldots \quad \ldots A_i \vdash \Phi (\text{if } b_i = 0) \ldots}{\vdash \Phi}\,\text{in}$$

If $a_j = 1$ in t_c, then A_j occurs as a lemma in the rule; if $a_i = 0$ in t_c, then A_i occurs as a case. The rules are given in abbreviated form and it should be understood that all judgments can be used with an extended hypotheses set Γ. So the elimination rule in full reads as follows (where Γ is a set of propositions).

$$\frac{\Gamma \vdash \Phi \quad \ldots \Gamma \vdash A_j \ (\text{if } a_j = 1) \ldots \quad \ldots \Gamma, A_i \vdash D \ (\text{if } a_i = 0) \ldots}{\Gamma \vdash D}\,\text{el}$$

In an elimination rule, we call $\vdash \Phi$ the *major premise* and the other hypotheses of the rule we call the *minor premises*.

▶ **Definition 2.** Given a set of connectives $\mathcal{C} := \{c_1, \ldots, c_n\}$, we define the *intuitionistic natural deduction system* for $\mathcal{C}$, $\mathsf{IPC}_{\mathcal{C}}$, by the following derivation rules.

- The *axiom rule*

$$\frac{}{\Gamma \vdash A}\,\text{axiom (if } A \in \Gamma)$$

- The elimination rules for the connectives in $\mathcal{C}$ and the intuitionistic introduction rules for the connectives in $\mathcal{C}$, as given in Definition 1.

We write $\Gamma \vdash_{\mathcal{C}} A$ if $\Gamma \vdash A$ is derivable using the derivation rules of $\mathsf{IPC}_{\mathcal{C}}$.

▶ **Example 3.**

A	B	$A \vee B$	$A \wedge B$	$A \to B$	$\neg A$
0	0	0	0	1	1
0	1	1	0	1	1
1	0	1	0	0	0
1	1	1	1	1	0

1. From the truth table for $\vee$ we derive the following intuitionistic rules for $\vee$. We label the rules by the relevant entries in the truth table.

$$\frac{\vdash A \vee B \qquad A \vdash D \qquad B \vdash D}{\vdash D}\vee\text{-el} \qquad \frac{A \vdash A \vee B \qquad \vdash B}{\vdash A \vee B}\vee\text{-in}_{01}$$

$$\frac{\vdash A \qquad B \vdash A \vee B}{\vdash A \vee B}\vee\text{-in}_{10} \qquad \frac{\vdash A \qquad \vdash B}{\vdash A \vee B}\vee\text{-in}_{11}$$

These rules are all intuitionistically correct, as one can observe by inspection. We will show that these are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 1 elimination rule and 2 introduction rules, which are the well-known ones.

2. From the truth table for $\wedge$ we derive the following intuitionistic rules for $\wedge$, 3 elimination rules and one introduction rule.

$$\frac{\vdash A \wedge B \qquad A \vdash D \qquad B \vdash D}{\vdash D}\wedge\text{-el}_{00} \qquad \frac{\vdash A \wedge B \qquad A \vdash D \qquad \vdash B}{\vdash D}\wedge\text{-el}_{01}$$

$$\frac{\vdash A \wedge B \qquad \vdash A \qquad B \vdash D}{\vdash D}\wedge\text{-el}_{10} \qquad \frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\wedge\text{-in}$$

These rules are all intuitionistically correct, as one can observe by inspection. We will show that these are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 2 elimination rules and 1 introduction rule, which are the well-known ones. The elimination rules for $\wedge$ have a bit the flavor of the so called "general elimination rules" of Schroeder-Heister [17] and Von Plato [23], in the sense that we don't derive A, respectively B, from $A \wedge B$, but an auxiliary conclusion D is derived. This rule, also called the *parallel elimination rule* by Tennant [21], is as follows.

$$\frac{\vdash A \wedge B \qquad A, B \vdash D}{\vdash D}\wedge\text{-el}^{\text{par}}$$

We will show that this rule can be derived from ours. See Definition 45 and Lemma 46, where this is shown using proof-terms.

3. From the truth table for $\neg$ we also derive the following rules for $\neg$, one elimination rule and one introduction rule.

$$\frac{A \vdash \neg A}{\vdash \neg A}\neg\text{-in} \qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash D}\neg\text{-el}$$

The elimination rule is familiar. For the introduction rule: to prove $\neg A$, one "only" has to prove $\neg A$ from A, which may seem limited. The traditional $\neg$-in rule is the following.

$$\frac{A \vdash \neg B \qquad A \vdash B}{\vdash \neg A}\neg\text{-in}^{\text{t}}$$

The two ¬-introduction rules are equivalent, which we will show in detail (using proof terms) in Lemma 53. To derive ¬-in$^{\mathrm{t}}$ from ¬-in one also needs ¬-el, so we view ¬-in as more primitive then the traditional rule ¬-in$^{\mathrm{t}}$.

As an example of the intuitionistic derivation rules for ¬ we show that $A \vdash \neg\neg A$ is derivable:

$$\dfrac{\dfrac{A, \neg A \vdash \neg A \qquad A, \neg A \vdash A}{A, \neg A \vdash \neg\neg A}\ \neg\text{-el}}{A \vdash \neg\neg A}\ \neg\text{-in}$$

4. From the truth table for → we derive the following intuitionistic rules for →.

$$\dfrac{A \vdash A \to B \qquad B \vdash A \to B}{\vdash A \to B}\ {\to}\text{-in}_{00} \qquad \dfrac{\vdash A \to B \qquad \vdash A \qquad B \vdash D}{\vdash D}\ {\to}\text{-el}$$

$$\dfrac{A \vdash A \to B \qquad \vdash B}{\vdash A \to B}\ {\to}\text{-in}_{10} \qquad \dfrac{\vdash A \qquad \vdash B}{\vdash A \to B}\ {\to}\text{-in}_{11}$$

These rules are all intuitionistically correct, as one can verify by inspection. For example, for →-in$_{01}$, observe that if $A \vdash A \to B$, then $\vdash A \to B$, so the second hypothesis is superfluous. Similarly for →-in$_{11}$, the first hypothesis is superfluous. We will show that these rules are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 1 elimination rule and 2 introduction rules. These are not the well-known ones, because the well-known →-in-rule does not fit into our scheme:

$$\dfrac{A \vdash B}{\vdash A \to B}\ {\to}\text{-in}$$

In this rule, both the conclusion is changed *and* an assumption (case) is added. In our system, each rule has the property that a hypothesis either adds an assumption or changes the conclusion (while retaining the same set of assumptions), and this "or" is exclusive.

We continue this section with some more basic properties and notions, most of which have been described briefly in [7]. We also introduce some further notation.

In the logic $\mathsf{IPC}_{\mathcal{C}}$ (Definitions 1 and 2) we can freely reuse formulas and weaken the context, so the structural rules of contraction and weakening are wired into the system. Because weakening is used a lot, we formulate it as a Lemma. The proof is an immediate induction on the derivation.

▶ **Lemma 4** (Weakening). *If* $\Gamma \vdash A$ *with derivation* Π *and* $\Gamma \subseteq \Delta$*, then* $\Delta \vdash A$ *with derivation* Π.

In natural deduction in tree format, the elimination of a detour convertibility involves *composition* of derivations: the placing of one derivation on top of another, replacing a discharged leaf A on top of a derivation tree (an assumption) by a derivation of A. In natural deduction in sequent calculus style, this amounts to replacing an axiom $\Gamma, A \vdash A$, that appears as the leaf of a derivation tree, by a derivation of $\Delta \vdash A$, where $\Delta \subset \Gamma$. We first define more precisely how the composition of derivation works in natural deduction in sequent calculus style.

▶ **Lemma 5.** *If* $\Delta, \varphi \vdash \psi$*, and* $\Gamma \vdash \varphi$*, then* $\Gamma, \Delta \vdash \psi$

Proof. By induction on the derivation of $\Delta, \varphi \vdash \psi$, using weakening (Lemma 4). ◀

To be a bit more precise about what happens with the derivations in the proof of Lemma 5, let Π be the derivation of $\Delta, \varphi \vdash \psi$. Then, due to the format of our rules:

- The only place in Π where the hypothesis φ is actually used is at a leaf of Π, in an instance of the (axiom) rule.
- Contexts can only grow when we walk upwards in a derivation, so these leaves are of the form $\Delta', \varphi \vdash \varphi$ for some $\Delta' \supseteq \Delta$.

We replace this leaf by Σ, the derivation of $\Gamma \vdash \varphi$. Due to weakening, this Σ is also a derivation of $\Gamma, \Delta' \vdash \varphi$, so Π with the leaves of the form $\Delta', \varphi \vdash \varphi$ replaced by Σ yields a correct derivation of $\Gamma, \Delta \vdash \psi$.

▶ Notation 6. If Π is a derivation of $\Delta, \varphi \vdash \psi$ and Σ is a derivation of $\Gamma \vdash \varphi$, then we have a derivation of $\Gamma, \Delta \vdash \psi$ that looks like this:

$$\begin{array}{c} \begin{array}{ccc} \vdots\,\Sigma & & \vdots\,\Sigma \\ \Gamma \vdash \varphi & \ldots & \Gamma \vdash \varphi \end{array} \\ \vdots\,\Pi \\ \Gamma, \Delta \vdash \psi \end{array}$$

So in Π, every application of an (axiom) rule at a leaf, deriving $\Delta' \vdash \varphi$ for some $\Delta' \supseteq \Delta$ is replaced by a copy of a derivation Σ, which is also a derivation of $\Delta', \Gamma \vdash \varphi$.

The fact that we have weakening supports the following convention.

▶ Convention 7. In examples, to simplify derivations we will often use the following format for an elimination rule (and similarly for an introduction rule).

$$\frac{\Gamma_0 \vdash \Phi \qquad \ldots \Gamma_j \vdash A_j \ (\text{if } a_j = 1) \ldots \qquad \ldots \Gamma_i, A_i \vdash D \ (\text{if } a_i = 0) \ldots}{\cup_{k=0}^{n} \Gamma_k \vdash D} \text{el}$$

This prevents us from having to copy the full Γ from the conclusion to the hypotheses in a rule; we can limit ourselves to the parts of Γ that we need for that particular branch in the derivation.

We now recall from [7] two lemmas that allow to reduce the number of deduction rules: some rules can be taken together and one or more of the hypotheses can be dropped. For completeness, we give these lemmas again here (Lemma 9 and Lemma 12), with their proofs. First, we motivate Lemma 9 by looking at the example of the rules for $\wedge$ (Example 3).

▶ **Example 8.** From the truth table we have derived the following 3 intuitionistic elimination rules for $\wedge$.

$$\frac{\vdash A \wedge B \qquad A \vdash D \qquad B \vdash D}{\vdash D} \wedge\text{-el}_{00} \qquad \frac{\vdash A \wedge B \qquad A \vdash D \qquad \vdash B}{\vdash D} \wedge\text{-el}_{01}$$

$$\frac{\vdash A \wedge B \qquad \vdash A \qquad B \vdash D}{\vdash D} \wedge\text{-el}_{10}$$

These rules can be reduced to the following 2 equivalent elimination rules. The index in the rule indicates where it originates from: $\wedge$-el$_{0_}$ is the combination of $\wedge$-el$_{00}$ and $\wedge$-el$_{01}$.

$$\frac{\vdash A \wedge B \qquad A \vdash D}{\vdash D} \wedge\text{-el}_{0\_} \qquad \frac{\vdash A \wedge B \qquad B \vdash D}{\vdash D} \wedge\text{-el}_{\_0}$$

It can be shown that these sets of rules are equivalent. Here we only show the derivability of the $\wedge$-el$_{0_}$ rule from the rules $\wedge$-el$_{00}$ and $\wedge$-el$_{01}$. As usual, for notational simplicity we suppress the context Γ. Suppose we have derivations of $\vdash A \wedge B$ and of $A \vdash D$. Then we have the following derivation.

$$\dfrac{\vdash A \wedge B \qquad A \vdash D \qquad \dfrac{B \vdash A \wedge B \qquad B, A \vdash D \qquad B \vdash B}{B \vdash D}\,\wedge\text{-el}_{01}}{\vdash D}\,\wedge\text{-el}_{00}$$

Note that the third and fourth hypothesis come from the first and second through weakening, and the fifth hypothesis is the axiom rule

The general method here is that we can replace two rules that only differ in one hypothesis, which in one rule occurs as a lemma and in the other as a case, by one rule where the hypothesis is removed. It will be clear that the Γ's above are not relevant for the argument, so we will not write these.

▶ **Lemma 9.** *A system with two derivation rules of the form*

$$\dfrac{\vdash A_1 \ \ldots \ \vdash A_n \quad B_1 \vdash D \ \ldots \ B_m \vdash D \quad A \vdash D}{\vdash D} \qquad \dfrac{\vdash A_1 \ \ldots \ \vdash A_n \quad \vdash A \quad B_1 \vdash D \ \ldots \ B_m \vdash D}{\vdash D}$$

is equivalent to the system with these two rules replaced by

$$\dfrac{\vdash A_1 \ \ldots \ \vdash A_n \qquad B_1 \vdash D \ \ldots \ B_m \vdash D}{\vdash D}$$

Proof. The implication from bottom to top is immediate. From top to bottom, suppose we have the two given rules. We now derive the bottom one. Assume we have derivations of $\vdash A_1, \ldots, \vdash A_n, B_1 \vdash D, \ldots, B_m \vdash D$. We now have the following derivation of $\vdash D$.

$$\dfrac{\vdash A_1 \ \ldots \ \vdash A_n \quad B_1 \vdash D \ \ldots \ B_m \vdash D \quad \dfrac{A \vdash A_1 \ \ldots \ A \vdash A_n \quad A \vdash A \quad A, B_1 \vdash D \ \ldots \ A, B_m \vdash D}{A \vdash D}}{\vdash D}$$

◀

Lemma 9 can be applied to elimination and introduction rules. An application to elimination rules is given in Example 8. We now give two applications to introduction rules.

▶ **Example 10.** From the truth table we have derived the following 3 intuitionistic introduction rules for ∨.

$$\dfrac{A \vdash A \vee B \qquad \vdash B}{\vdash A \vee B}\,\vee\text{-in}_{01} \qquad \dfrac{\vdash A \qquad B \vdash A \vee B}{\vdash A \vee B}\,\vee\text{-in}_{10} \qquad \dfrac{\vdash A \qquad \vdash B}{\vdash A \vee B}\,\vee\text{-in}_{11}$$

Using Lemma 9, these rules can be reduced to the following 2 equivalent introduction rules. (We could call ∨-inl also ∨-in_1, but we use a more informative and standard name: "in-left".)

$$\dfrac{\vdash A}{\vdash A \vee B}\,\vee\text{-inl} \qquad \dfrac{\vdash B}{\vdash A \vee B}\,\vee\text{-inr}$$

▶ **Example 11.** Similar to ∨, we can optimize the introduction rules for →. From the truth table we have derived the following 3 intuitionistic introduction rules for →.

$$\dfrac{A \vdash A \to B \qquad B \vdash A \to B}{\vdash A \to B}\,\to\text{-in}_{00} \qquad \dfrac{A \vdash A \to B \qquad \vdash B}{\vdash A \to B}\,\to\text{-in}_{01} \qquad \dfrac{\vdash A \qquad \vdash B}{\vdash A \to B}\,\to\text{-in}_{11}$$

Using Lemma 9, these rules can be reduced to the following 2 equivalent introduction rules.

$$\frac{A \vdash A \to B}{\vdash A \to B}\to\text{-in}_a \qquad \frac{\vdash B}{\vdash A \to B}\to\text{-in}_b$$

It can easily be shown that the rules $\to$-in$_a$ and $\to$-in$_b$ together are equivalent with the well-known $\to$-in:

$$\frac{A \vdash B}{\vdash A \to B}\to\text{-in}$$

NB. To derive $\to$-in$_a$ from $\to$-in, one also needs $\to$-el.

As $\to$-in does not conform with our format for rules, we will be using $\to$-in$_a$ and $\to$-in$_b$ as our basic rules and treat $\to$-in as a defined rule, the composition of first $\to$-in$_b$ and then $\to$-in$_a$.

Another optimization we can perform is to replace a rule which has only one case by a rule where the case is the conclusion. To illustrate this: the simplified elimination rules for $\wedge$, $\wedge$-el$_{0_}$ and $\wedge$-el$__0$ have only one case. The rule $\wedge$-el$_{0_}$ can thus be replaced by the rule $\wedge$-ell, which is the usual left projection rule, $\wedge$-elimination-left.

$$\frac{\vdash A \wedge B \qquad A \vdash D}{\vdash D}\wedge\text{-el}_{0\_} \qquad \frac{\vdash A \wedge B}{\vdash A}\wedge\text{-ell}$$

There is a general Lemma stating this simplification is correct.

▶ **Lemma 12.** *A system with a derivation rule of the form to the left is equivalent to the system with this rule replaced by the rule on the right.*

$$\frac{\vdash A_1 \;\ldots\; \vdash A_n \qquad B \vdash D}{\vdash D} \qquad\qquad \frac{\vdash A_1 \;\ldots\; \vdash A_n}{\vdash B}$$

Proof. The implication from left to right is immediate. From right to left, assume we have derivations of $\vdash A_1, \ldots, \vdash A_n$. Then, by the rule to the right, we have $\Gamma \vdash B$. Now assume we also have a derivation of $B \vdash D$. By Lemma 5, we also have a derivation of $\Gamma \vdash D$. ◀

▶ **Definition 13.** The *standard derivation rules* for the intuitionistic propositional connectives $\wedge, \vee, \to, \neg, \bot$ and $\top$ are given below. These rules are derived from the truth tables and optimized following Lemmas 9 and 12. We have seen most of the rules in previous Examples, except for the rules for $\top$ and $\bot$, which are derived immediately from Definition 1. The system with these connectives and rules we will call *intuitionistic proposition logic* and if we want to explicit we write $\Gamma \vdash_i A$ for derivability in this system.

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\wedge\text{-in} \qquad \frac{\vdash A \wedge B}{\vdash A}\wedge\text{-ell} \qquad \frac{\vdash A \wedge B}{\vdash B}\wedge\text{-elr}$$

$$\frac{\vdash A}{\vdash A \vee B}\vee\text{-inl} \qquad \frac{\vdash B}{\vdash A \vee B}\vee\text{-inr} \qquad \frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\vee\text{-el}$$

$$\frac{A \vdash A \to B}{\vdash A \to B}\to\text{-in}_a \qquad \frac{\vdash B}{\vdash A \to B}\to\text{-in}_b \qquad \frac{\vdash A \to B \quad \vdash A}{\vdash B}\to\text{-el}$$

$$\frac{A \vdash \neg A}{\vdash \neg A}\neg\text{-in} \qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash D}\neg\text{-el} \qquad \frac{}{\vdash \top}\top\text{-in} \qquad \frac{\vdash \bot}{\vdash D}\bot\text{-el}$$

2.1 Three larger examples

As examples we look in more detail at two ternary connectives and one binary connective. The ternary connectives we treat are $\mathsf{if\text{-}then\text{-}else}$, the “if-then-else” connective, and most, the ternary connective that is true if at least 2 of the arguments are true. These have been discussed in finer detail in [7], notably the connective $\mathsf{if\text{-}then\text{-}else}$. The binary connective that we study at the end of this section is the nand, written $A \uparrow B$ for $\mathsf{nand}(A, B)$. It is also known as the *Sheffer stroke*, the well-known connective that is functionally complete classically, where $A \uparrow B$ expresses $\neg(A \wedge B)$.

The truth tables of most and $\mathsf{if\text{-}then\text{-}else}$ are as follows, where we denote $\mathsf{if}\ A\ \mathsf{then}\ B\ \mathsf{else}\ C$ by $A{\to}B/C$.

A	B	C	$\mathsf{most}(A, B, C)$	$A{\to}B/C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

From the lines in the truth table of $A{\to}B/C$ with a 0 we get the following four elimination rules.

$$\frac{\vdash A{\to}B/C \quad A \vdash D \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad A \vdash D \quad \vdash B \quad C \vdash D}{\vdash D}$$

$$\frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad \vdash C}{\vdash D}$$

Using Lemmas 9 and 12, these can be reduced to the following two. (The two rules on the first line reduce to else-el, the two rules on the second line reduce to then-el.)

$$\boxed{\frac{\vdash A{\to}B/C \quad A \vdash D \quad C \vdash D}{\vdash D}\ \text{else-el} \qquad \frac{\vdash A{\to}B/C \quad \vdash A}{\vdash B}\ \text{then-el}}$$

These are not the only possible optimizations: the two rules on the left can also be combined into an “if-el” rule:

$$\frac{\vdash A{\to}B/C \quad B \vdash D \quad C \vdash D}{\vdash D}\ \text{if-el}$$

From the lines in the truth table of $A{\to}B/C$ with a 1 we get the following four introduction rules:

$$\frac{A \vdash A{\to}B/C \quad B \vdash A{\to}B/C \quad \vdash C}{\vdash A{\to}B/C} \qquad \frac{A \vdash A{\to}B/C \quad \vdash B \quad \vdash C}{\vdash A{\to}B/C}$$

$$\frac{\vdash A \quad \vdash B \quad C \vdash A{\to}B/C}{\vdash A{\to}B/C} \qquad \frac{\vdash A \quad \vdash B \quad \vdash C}{\vdash A{\to}B/C}$$

Using Lemmas 9 and 12 can be reduced to the following two. (The two rules on the first line reduce to else-in, the two rules on the second line reduce to then-in.)

$$\frac{A \vdash A{\to}B/C \quad \vdash C}{\vdash A{\to}B/C}\ \text{else-in} \qquad \frac{\vdash A \quad \vdash B}{\vdash A{\to}B/C}\ \text{then-in}$$

Again, these are not the only possible optimizations: the two rules on the right can also be combined into an "if-in" rule:

$$\frac{\vdash B \quad \vdash C}{\vdash A{\to}B/C}\ \text{if-in}$$

In [7], we have studied the if-then-else connective in more detail, and we have shown that if-then-else, together with $\top$ and $\bot$ is *functionally complete*: all other constructive connectives can be defined in terms of it.

From the lines in the truth table of $\mathsf{most}(A, B, C)$ with a 0 we get the following four elimination rules.

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D \quad \vdash C}{\vdash D}$$

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad \vdash B \quad C \vdash D}{\vdash D} \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad \vdash A \quad B \vdash D \quad C \vdash D}{\vdash D}$$

Using Lemmas 9 and 12, these can be reduced to the following three. If we would follow the naming conventions that we introduced earlier, we would have $\mathsf{most\text{-}el}_1 = \mathsf{most\text{-}el}_{00\_}$, $\mathsf{most\text{-}el}_2 = \mathsf{most\text{-}el}_{0\_0}$ and $\mathsf{most\text{-}el}_3 = \mathsf{most\text{-}el}_{\_00}$, but we will not pursue that naming here.

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D}{\vdash D}\ \mathsf{most\text{-}el}_1 \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad C \vdash D}{\vdash B}\ \mathsf{most\text{-}el}_2$$

$$\frac{\vdash \mathsf{most}(A,B,C) \quad B \vdash D \quad C \vdash D}{\vdash B}\ \mathsf{most\text{-}el}_3$$

From the lines in the truth table of $\mathsf{most}(A, B, C)$ with a 1 we get the following four introduction rules:

$$\frac{A \vdash \mathsf{most}(A,B,C) \quad \vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)} \qquad \frac{\vdash A \quad B \vdash \mathsf{most}(A,B,C) \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}$$

$$\frac{\vdash A \quad \vdash B \quad C \vdash \mathsf{most}(A,B,C)}{\vdash \mathsf{most}(A,B,C)} \qquad \frac{\vdash A \quad \vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}$$

Using Lemmas 9 and 12 can be reduced to the following three.

$$\frac{\vdash A \quad \vdash B}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most\text{-}in}_1 \qquad \frac{\vdash A \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most\text{-}in}_2 \qquad \frac{\vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most\text{-}in}_3$$

The truth table for $\mathsf{nand}(A, B)$, which we write as $A \uparrow B$ is as follows.

A	B	$A \uparrow B$
0	0	1
0	1	1
1	0	1
1	1	0

From this we derive the following 3 introduction and 1 elimination rule

$$\frac{A \vdash A \uparrow B \qquad B \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-in}_{00} \qquad \frac{A \vdash A \uparrow B \qquad \vdash B}{\vdash A \uparrow B}\uparrow\text{-in}_{01}$$

$$\frac{\vdash A \qquad B \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-in}_{10} \qquad \frac{\vdash A \uparrow B \qquad \vdash A \qquad \vdash B}{\vdash D}\uparrow\text{-el}$$

The three introduction rules can be combined to two rules, so our optimized set of deduction rules for nand consists of three rules. We call this nand-logic.

▶ **Definition 14.** The logic with just the connective nand and the three derivation rules below we define as nand-*logic*. We denote derivability in this logic by $\Gamma \vdash_\uparrow A$.

$$\frac{A \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-inl} \qquad \frac{B \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-inr} \qquad \frac{\vdash A \uparrow B \quad \vdash A \quad \vdash B}{\vdash D}\uparrow\text{-el}$$

We can define the usual connectives of intuitionistic proposition logic (Definition 13) in terms of nand in the usual way. This gives rise to an embedding of intuitionistic proposition logic into the nand-logic.

▶ **Definition 15.**

$$\begin{aligned}
\dot{\neg} A &:= A \uparrow A \\
A \mathbin{\dot\vee} B &:= (A \uparrow A) \uparrow (B \uparrow B) \\
A \mathbin{\dot\wedge} B &:= (A \uparrow B) \uparrow (A \uparrow B) \\
A \mathbin{\dot\to} B &:= A \uparrow (B \uparrow B)
\end{aligned}$$

This gives rise to the following interpretation of intuitionistic proposition logic into nand-logic.

$$\begin{aligned}
p^\uparrow &:= \dot{\neg}\dot{\neg} p \text{ for } p \text{ proposition letter} \\
(\neg A)^\uparrow &:= \dot{\neg} A^\uparrow \\
(A \vee B)^\uparrow &:= A^\uparrow \mathbin{\dot\vee} B^\uparrow \\
(A \wedge B)^\uparrow &:= A^\uparrow \mathbin{\dot\wedge} B^\uparrow \\
(A \to B)^\uparrow &:= A^\uparrow \mathbin{\dot\to} B^\uparrow
\end{aligned}$$

This interpretation extends straightforwardly to sets of propositions.

As a side remark, the translation of a proposition letter p could also be chosen to be p in stead of $\dot{\neg}\dot{\neg} p$. Then the soundness statement below (Proposition 17) requires an additional double negation: If $\Gamma \vdash_i A$, then $\Gamma^\uparrow \vdash_\uparrow \dot{\neg}\dot{\neg} A^\uparrow$. The connective $\uparrow$ is very much a "negative connective" and the choice of $\dot{\neg}\dot{\neg} p$ as translation of p renders all formulas $A^\uparrow$ negative, so the double negation can be avoided.

Before proving the soundness of the interpretation we give some auxiliary lemmas.

▶ **Lemma 16.** *In* nand*-logic, we have the following.*

1. *For arbitrary propositions A and B,*

$$\dot{\neg}\dot{\neg}(A \uparrow B) \vdash A \uparrow B,$$

2. *For every A,*

$$\dot{\neg}\dot{\neg}\dot{\neg} A \vdash \dot{\neg} A.$$

3. *For every proposition P from intuitionistic proposition logic,*

$$\dot\neg\dot\neg P^{\uparrow} \vdash P^{\uparrow}.$$

4. *For arbitrary propositions A and B,*

$$\textit{If } \Gamma, A \vdash B \textit{ then } \Gamma, \dot\neg B \vdash \dot\neg A.$$

Proof. The following proves $\dot\neg\dot\neg(A \uparrow B) \vdash A \uparrow B$. Here $\Gamma = \dot\neg\dot\neg(A \uparrow B), A, B, A \uparrow B$ and the last $\uparrow$-in rule denotes a successive application of $\uparrow$-inl followed by $\uparrow$-inr. Finally, the lowest $\uparrow$-el has one premise more, which is an exact copy of the derivation of $\dot\neg\dot\neg(A \uparrow B), A, B \vdash \dot\neg(A \uparrow B)$ that is given.

$$\dfrac{\dfrac{\dot\neg\dot\neg(A \uparrow B), A, B \vdash \dot\neg\dot\neg(A \uparrow B) \qquad \dfrac{\dfrac{\Gamma \vdash A \uparrow B \qquad \Gamma \vdash A \qquad \Gamma \vdash B}{\dot\neg\dot\neg(A \uparrow B), A, B, A \uparrow B \vdash \dot\neg(A \uparrow B)}\,\uparrow\text{-el}}{\dot\neg\dot\neg(A \uparrow B), A, B \vdash \dot\neg(A \uparrow B)}\,\uparrow\text{-in}}{\dot\neg\dot\neg(A \uparrow B), A, B \vdash A \uparrow B}\,\uparrow\text{-el}}{\dot\neg\dot\neg(A \uparrow B) \vdash A \uparrow B}\,\uparrow\text{-in}$$

So, $\dot\neg\dot\neg\dot\neg A \vdash \dot\neg A$ follows immediately, and similarly $\dot\neg\dot\neg P^{\uparrow} \vdash P^{\uparrow}$ for every proposition P from intuitionistic proposition logic.

Now, assuming that $\Gamma, A \vdash B$, we can make the following derivation of $\Gamma, \dot\neg B \vdash \dot\neg A$, using the fact that $\Gamma, B \uparrow B, A \vdash B$ by weakening.

$$\dfrac{\dfrac{\Gamma, B \uparrow B, A \vdash B \uparrow B \qquad \Gamma, B \uparrow B, A \vdash B \qquad \Gamma, B \uparrow B, A \vdash B}{\Gamma, B \uparrow B, A \vdash A \uparrow A}\,\uparrow\text{-el}}{\Gamma, B \uparrow B \vdash A \uparrow A}\,\uparrow\text{-in}$$

◀

We can now prove the soundness of the interpretation of intuitionistic proposition logic into nand-logic.

▶ Proposition 17. If $\Gamma \vdash_i A$, then $\Gamma^{\uparrow} \vdash_{\uparrow} A^{\uparrow}$.

Proof. The proof is by induction on the derivation of $\Gamma \vdash_i A$, so we have to show that the rules of intuitionistic proposition logic are sound inside nand-logic (after interpretation). We use Lemma 16, notably case (4), which we indicate explicitly in the derivations.

- $\neg$-in: we show that $\neg$-in of Definition 13 is derivable.

$$\dfrac{A \vdash A \uparrow A}{A \vdash A \uparrow A}\,\uparrow\text{-in}$$

- $\neg$-el: we show that $\neg$-el of Definition 13 is derivable.

$$\dfrac{\vdash A \uparrow A \qquad \vdash A \qquad \vdash A}{\vdash D}\,\uparrow\text{-el}$$

- $\vee$-in: we show that $A \vdash_{\uparrow} A \dot\vee B$ is derivable.

$$\dfrac{\dfrac{A, A \uparrow A \vdash A \uparrow A \qquad A, A \uparrow A \vdash A \qquad A, A \uparrow A \vdash A}{A, A \uparrow A \vdash (A \uparrow A) \uparrow (B \uparrow B)}\,\uparrow\text{-el}}{A \vdash (A \uparrow A) \uparrow (B \uparrow B)}\,\uparrow\text{-inl}$$

- $\vee$-el: we show that the following rule is derivable (which suffices).

$$\frac{\vdash A \mathbin{\dot\vee} B \qquad A \vdash D \qquad B \vdash D}{\vdash \dot\neg\dot\neg D}$$

$$\dfrac{\dfrac{\vdash (A \uparrow A) \uparrow (B \uparrow B) \qquad \dfrac{A \vdash D}{D \uparrow D \vdash A \uparrow A}\,16(4) \qquad \dfrac{B \vdash D}{D \uparrow D \vdash B \uparrow B}\,16(4)}{D \uparrow D \vdash (D \uparrow D) \uparrow (D \uparrow D)}\,\uparrow\text{-el}}{\vdash (D \uparrow D) \uparrow (D \uparrow D)}\,\uparrow\text{-inl}$$

- $\wedge$-el: we show that $A \mathbin{\dot\wedge} B \vdash_{\uparrow} \dot\neg\dot\neg A$ is derivable.

$$\dfrac{\dfrac{\dfrac{A \mathbin{\dot\wedge} B \vdash (A \uparrow B) \uparrow (A \uparrow B) \qquad \dfrac{\dfrac{A \uparrow A \vdash A \uparrow A \quad A \vdash A}{A \uparrow A, A \vdash A \uparrow B}\,\uparrow\text{-el}}{A \uparrow A \vdash A \uparrow B}\,\uparrow\text{-inl}}{A \mathbin{\dot\wedge} B, A \uparrow A \vdash A}\,\uparrow\text{-el}}{A \mathbin{\dot\wedge} B, A \uparrow A \vdash (A \uparrow A) \uparrow (A \uparrow A)}\,16(4)}{A \mathbin{\dot\wedge} B \vdash (A \uparrow A) \uparrow (A \uparrow A)}\,\uparrow\text{-inl}$$

- $\wedge$-in: we show that the following rule is derivable (which suffices).

$$\frac{\vdash A \qquad \vdash B}{\vdash A \mathbin{\dot\wedge} B}$$

$$\dfrac{\dfrac{A \uparrow B \vdash A \uparrow B \qquad \vdash A \qquad \vdash B}{A \uparrow B \vdash (A \uparrow B) \uparrow (A \uparrow B)}\,\uparrow\text{-el}}{\vdash (A \uparrow B) \uparrow (A \uparrow B)}\,\uparrow\text{-inl}$$

- $\to$-in: we show that the following rule is derivable (which suffices).

$$\frac{A \vdash B}{\vdash A \mathbin{\dot\to} B}$$

$$\dfrac{\dfrac{\dfrac{B \uparrow B \vdash B \uparrow B \qquad A \vdash B \qquad A \vdash B}{A, B \uparrow B \vdash A \uparrow (B \uparrow B)}\,\uparrow\text{-el}}{A \vdash A \uparrow (B \uparrow B)}\,\uparrow\text{-inr}}{\vdash A \uparrow (B \uparrow B)}\,\uparrow\text{-inl}$$

- $\to$-el: we show that the following rule is derivable (which suffices).

$$\frac{\vdash A \mathbin{\dot\to} B \qquad \vdash A}{\vdash \dot\neg\dot\neg B}$$

$$\dfrac{\dfrac{\dfrac{\vdash A \uparrow (B \uparrow B) \qquad \vdash A \qquad B \uparrow B \vdash B \uparrow B}{B \uparrow B \vdash B}\,\uparrow\text{-el}}{B \uparrow B \vdash (B \uparrow B) \uparrow (B \uparrow B)}\,16(4)}{\vdash (B \uparrow B) \uparrow (B \uparrow B)}\,\uparrow\text{-inl}$$

◀

The reverse of Proposition 17 does not hold. For example, $\not\vdash p \vee \neg p$, for p a proposition letter, while $(p \vee \neg p)^{\uparrow} = (\dot{p} \uparrow \dot{p}) \uparrow (\dot{\neg}\dot{p} \uparrow \dot{\neg}\dot{p})$, where $\dot{p} := \dot{\neg}\dot{\neg}p$. The proposition $(A \uparrow A) \uparrow (\dot{\neg}A \uparrow \dot{\neg}A)$ is derivable in nand-logic for any A (note that $\dot{\neg}A = A \uparrow A$):

$$\dfrac{\dfrac{\dot{\neg}A \uparrow \dot{\neg}A \vdash \dot{\neg}A \uparrow \dot{\neg}A \qquad A \uparrow A \vdash \dot{\neg}A \qquad A \uparrow A \vdash \dot{\neg}A}{A \uparrow A, \dot{\neg}A \uparrow \dot{\neg}A \vdash (A \uparrow A) \uparrow (\dot{\neg}A \uparrow \dot{\neg}A)}\,\uparrow\text{-el}}{\vdash (A \uparrow A) \uparrow (\dot{\neg}A \uparrow \dot{\neg}A)}\,\uparrow\text{-in}$$

There is also an obvious mapping from nand-logic to intuitionistic proposition logic, by interpreting $A \uparrow B$ as $\neg(A \wedge B)$. As a matter of fact, it can also be shown in the joint system (i.e. where we add nand to intuitionistic proposition logic) that $A \uparrow B$ and $\neg(A \wedge B)$ are equivalent: $A \uparrow B \vdash \neg(A \wedge B)$ and $\neg(A \wedge B) \vdash A \uparrow B$. In presence of the implication and conjunction connective, the latter can be reformulated as $\vdash A \uparrow B \longleftrightarrow \neg(A \wedge B)$ (where, as usual, we let $C \longleftrightarrow D$ abbreviate $(C \to D) \wedge (D \to C)$).

▶ **Definition 18.** We define the mapping $(-)^{\downarrow}$ from nand-logic to intuitionistic proposition logic by defining

$$(A \uparrow B)^{\downarrow} := \neg(A^{\downarrow} \wedge B^{\downarrow})$$

and further by induction on propositions. This mapping extends to sets of hypotheses Γ in the obvious way.

▶ Proposition 19. If $\Gamma \vdash_{\uparrow} A$, then $\Gamma^{\downarrow} \vdash_i A^{\downarrow}$.

Proof. By induction on the derivation. The only thing to show is that the rules $\uparrow$-el, $\uparrow$-inl and $\uparrow$-inr are sound in intuitionistic proposition logic is we interpret $A \uparrow B$ as $\neg(A \wedge B)$. So we have to verify the soundness of the following rules.

$$\dfrac{A \vdash \neg(A \wedge B)}{\vdash \neg(A \wedge B)} \qquad \dfrac{B \vdash \neg(A \wedge B)}{\vdash \neg(A \wedge B)} \qquad \dfrac{\vdash \neg(A \wedge B) \quad \vdash A \quad \vdash B}{\vdash D}$$

A simple inspection shows that these rules are sound in intuitionistic proposition logic. ◀

We can now formulate a Glivenko-like theorem that relates nand-logic and intuitionistic proposition logic. (Glivenko's theorem, e.g. see [22], relates intuitionistic and classical proposition logic via the double negation.)

▶ Proposition 20. For A a proposition of intuitionistic proposition logic,

$$\vdash_i A^{\uparrow\downarrow} \longleftrightarrow \neg\neg A$$

.

Proof. By induction on the structure of A.

- $A = p$, a proposition letter. Then $p^{\uparrow\downarrow} = (\dot{\neg}\dot{\neg}p)^{\downarrow} = \neg(\neg(p \wedge p) \wedge \neg(p \wedge p)) \longleftrightarrow \neg\neg p$.
- $A = \neg B$. Then $(\neg B)^{\uparrow\downarrow} = (B \uparrow B)^{\downarrow} = \neg(B \wedge B) \longleftrightarrow \neg\neg\neg B$.
- $A = B \vee C$. Then $(B \vee C)^{\uparrow\downarrow} = ((B \uparrow B) \uparrow (C \uparrow C))^{\downarrow} = \neg(\neg(B \wedge B) \wedge \neg(C \wedge C)) \longleftrightarrow \neg\neg(B \vee C)$.
  For the equivalence $\neg(\neg B \wedge \neg C) \longleftrightarrow \neg\neg(B \vee C)$: from left to right, if $\neg(B \vee C)$, then $\neg B$ and $\neg C$, so we have a contradiction with $\neg(\neg B \wedge \neg C)$; from right to left, if $\neg B \wedge \neg C$, then $\neg B$ and so from $B \vee C$ we derive C, contradiction, so we derive $\neg(B \vee C)$, but this contradicts $\neg\neg(B \vee C)$, so we conclude that $\neg(\neg B \wedge \neg C)$

- $A = B \wedge C$. Then $(B \wedge C)^{\uparrow\downarrow} = ((B \uparrow C) \uparrow (B \uparrow C))^{\downarrow} = \neg(\neg(B \wedge C) \wedge \neg(B \wedge C)) \longleftrightarrow \neg\neg(B \wedge C)$.
- $A = B \to C$. Then $(B \to C)^{\uparrow\downarrow} = (B \uparrow (C \uparrow C))^{\downarrow} = \neg(B \wedge \neg(C \wedge C)) \longleftrightarrow \neg\neg(B \to C)$. For the equivalence $\neg(B \wedge \neg C) \longleftrightarrow \neg\neg(B \to C)$: From left to right, assume $\neg(B \to C)$; if C, then $B \to C$, so from $\neg(B \to C)$ we get $\neg C$; then if B we also have $B \wedge \neg C$, contradicting $\neg(B \wedge \neg C)$, so we have $\neg B$; but from $\neg B$ we get $B \to C$. Contradiction, so we conclude $\neg\neg(B \to C)$. From right to left: Assume $B \wedge \neg C$. Then $B \to C$ implies C, contradiction, so $\neg(B \to C)$, contradicting $\neg(B \to C)$, so we conclude $\neg(B \wedge \neg C)$.

◀

▶ **Corollary 21.** *For A a proposition in intuitionistic proposition logic,*

$$\vdash_i \neg\neg A \qquad \Longleftrightarrow \qquad \vdash_\uparrow A^\uparrow.$$

Proof. If $\vdash_i \neg\neg A$, then $\vdash_\uparrow \dot\neg\dot\neg A^\uparrow$ by Proposition 17, and so $\vdash_\uparrow \dot\neg\dot\neg A^\uparrow$ by Lemma 16(1).
If $\vdash_\uparrow A^\uparrow$, then $\vdash_i A^{\uparrow\downarrow}$ by Proposition 19, so $\vdash_i \neg\neg A$ by Proposition 20. ◀

3 Convertibilities and conversion

The notion of *detour convertibility* has already been described in [7]: an introduction of Φ immediately followed by an elimination of Φ. (In [7] it was called *direct cut* but – although the literature is not completely consistent on this point – the notion of cut is usually reserved for sequent calculus and for natural deduction one uses the terminology of convertibility.) In such case there is (referring back to the truth table, see Definition 1) at least one k for which $a_k \neq b_k$. In case $a_k = 0, b_k = 1$, we have a sub-derivation Σ of $\vdash A_k$ and a sub-derivation Θ of $A_k \vdash D$ and we can plug Σ on top of Θ to obtain a derivation of $\vdash D$. In case $a_k = 1, b_k = 0$, we have a sub-derivation Σ of $A_k \vdash \Phi$ and a sub-derivation Θ of $\vdash A_k$ and we can plug Θ on top of Σ to obtain a derivation of $\vdash \Phi$. This is then used as a hypothesis for the elimination rule (that remains in this case) instead of the original one that was a consequence of the introduction rule (that now disappears).

In general there are more k for which $a_k \neq b_k$, so the general detour conversion procedure is non-deterministic. We view this non-determinism as a natural feature in natural deduction; the fact that for some connectives (or combination of connectives), detour conversion is deterministic is an "emerging" property. We will show examples of the non-determinism of detour conversion later.

The introduction of a formula Φ immediately followed by an elimination of Φ we will call a *detour convertibility*. In general in between the introduction rule for Φ and the elimination rule for Φ, there may be other auxiliary rules, so occasionally we may have to first permute the elimination rule with these auxiliary rules to obtain a detour convertibility that can be reduced away. So, we will also define the notion of *permutation convertibility* and of *permutation conversion*.

▶ **Definition 22.** Let c be a connective of arity n, with an elimination rule and an intuitionistic introduction rule derived from the truth table, as in Definition 1. So suppose we have the following rules in the truth table t_c.

p_1	$\dots$	p_n	$c(p_1, \dots, p_n)$
a_1	$\dots$	a_n	0
b_1	$\dots$	b_n	1

A *detour convertibility* in a derivation is a pattern of the following form, where $\Phi = c(A_1, \dots, A_n)$.

$$\dfrac{\dfrac{\cdots\quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\quad \cdots\qquad \cdots\quad \begin{array}{c}\boxed{\Sigma_i}\\ \Gamma, A_i\vdash\Phi\end{array}\quad \cdots}{\Gamma\vdash\Phi}\,\text{in}\qquad \cdots\quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma\vdash A_k\end{array}\quad \cdots\qquad \cdots\quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell\vdash D\end{array}\quad \cdots}{\Gamma\vdash D}\,\text{el}$$

- Here, in is an arbitrary introduction rule. In this rule, A_j ranges over all propositions where $b_j = 1$; A_i ranges over all propositions where $b_i = 0$,
- Here, el is an arbitrary elimination rule. In this rule, A_k ranges over all propositions where $a_k = 1$; A_ℓ over all propositions where $a_\ell = 0$,

A *detour conversion* is defined by replacing the derivation pattern above by

1. If $\ell = j$ for some ℓ, j (that is: $A_\ell = A_j$):

$$\begin{array}{c} \begin{array}{c}\vdots\,\boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\quad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\\ \vdots\,\boxed{\Pi_\ell}\\ \Gamma\vdash D\end{array}$$

2. If $k = i$ for some k, i (that is: $A_k = A_i$):

$$\dfrac{\begin{array}{c} \begin{array}{c}\vdots\,\boxed{\Pi_k}\\ \Gamma\vdash A_i\end{array}\quad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Pi_k}\\ \Gamma\vdash A_i\end{array}\\ \vdots\,\boxed{\Sigma_i}\\ \Gamma\vdash\Phi\end{array}\qquad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Pi_k}\\ \Gamma\vdash A_k\end{array}\quad \cdots\qquad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Pi_\ell}\\ \Gamma, A_\ell\vdash D\end{array}\quad \cdots}{\Gamma\vdash D}\,\text{el}$$

There may be several choices for the i and j in the previous definition, so detour elimination is non-deterministic in general. We give an example of most to illustrate this. For simplicity, we use the optimized rules.

▶ **Example 23.** Consider the following detour convertibility for most.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\boxed{\Sigma_1}\\ \Gamma\vdash A\end{array}\qquad \begin{array}{c}\vdots\,\boxed{\Sigma_2}\\ \Gamma\vdash B\end{array}}{\Gamma\vdash \mathsf{most}(A,B,C)}\,\mathsf{most\text{-}in}_1\qquad \begin{array}{c}\vdots\,\boxed{\Pi_1}\\ \Gamma, A\vdash D\end{array}\qquad \begin{array}{c}\vdots\,\boxed{\Pi_2}\\ \Gamma, B\vdash D\end{array}}{\Gamma\vdash D}\,\mathsf{most\text{-}el}_1$$

Here we can reduce to either one of the following derivations of $\Gamma \vdash D$, which shows that the detour conversion process is not Church-Rosser. (Of course, one could fix a choice, e.g. always take the first possible detour convertibility from the left, but that would be completely arbitrary.)

$$\begin{array}{c} \begin{array}{c}\vdots\,\boxed{\Sigma_1}\\ \Gamma\vdash A\end{array}\quad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Sigma_1}\\ \Gamma\vdash A\end{array}\\ \vdots\,\boxed{\Pi_1}\\ \Gamma\vdash D\end{array}\qquad\qquad \begin{array}{c} \begin{array}{c}\vdots\,\boxed{\Sigma_2}\\ \Gamma\vdash B\end{array}\quad \cdots\quad \begin{array}{c}\vdots\,\boxed{\Sigma_2}\\ \Gamma\vdash B\end{array}\\ \vdots\,\boxed{\Pi_2}\\ \Gamma\vdash D\end{array}$$

A more concrete example is the following.

$$\dfrac{\dfrac{\dfrac{A\wedge B\vdash A\wedge B}{A\wedge B\vdash A}\,\wedge\text{-ell}\qquad \dfrac{A\wedge B\vdash A\wedge B}{A\wedge B\vdash B}\,\wedge\text{-elr}}{A\wedge B\vdash \mathsf{most}(A,B,C)}\,\mathsf{most}\text{-in}_1 \qquad \dfrac{A\vdash A}{A\vdash A\vee B}\,\vee\text{-inl}\qquad \dfrac{B\vdash B}{B\vdash A\vee B}\,\vee\text{-inr}}{A\wedge B\vdash A\vee B}\,\mathsf{most}\text{-el}_1$$

This derivation can either be reduced to a derivation of $A\wedge B\vdash A\vee B$ via $A\wedge B\vdash A$ or via $A\wedge B\vdash B$.

It can happen that the introduction of a formula $\Phi = c(A_1,\ldots,A_n)$ is not followed directly by an elimination for c, but first by other elimination rules, where Φ acts as a minor premise. In that way, a detour convertibility can be "blocked" by other elimination rules. So, apart from the detour conversion elimination arising from an introduction rule immediately followed by an elimination, we have a notion of "hidden" or *permutation convertibility*, where we want to permute one elimination rule over another.

▶ **Example 24.**

$$\dfrac{\dfrac{\Gamma\vdash A\vee B\qquad \dfrac{\Gamma,A,C\vdash C\to D}{\Gamma,A\vdash C\to D}\,\to\text{-in}_a\qquad \Gamma,B\vdash C\to D}{\Gamma\vdash C\to D}\,\vee\text{-el}\qquad \Gamma\vdash C}{\Gamma\vdash D}\,\to\text{-el}$$

In this derivation, the detour convertibility arising from $\to$-in$_a$ followed by $\to$-el is blocked by the $\vee$-el rule where the major premise of the $\to$-el rule is a minor premise. This is a *permutation convertibility*, which can be contracted by permuting the $\to$-el rule over the $\vee$-el rule.

▶ **Definition 25.** Let c and c' be connectives of arity n and n', with elimination rules r and r' respectively, both derived from the truth table. A *permutation convertibility* in a derivation is a pattern of the following form, where $\Phi = c(B_1,\ldots,B_n)$, $\Psi = c'(A_1,\ldots,A_{n'})$.

$$\dfrac{\dfrac{\Gamma\vdash\Psi\ldots\quad \begin{array}{c}\vdots\ \boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\quad\ldots\quad\ldots\quad \begin{array}{c}\vdots\ \boxed{\Sigma_i}\\ \Gamma,A_i\vdash\Phi\end{array}\quad\ldots}{\Gamma\vdash\Phi}\,\mathrm{el}_{r'}\qquad \ldots\quad \begin{array}{c}\vdots\ \boxed{\Pi_k}\\ \Gamma\vdash B_k\end{array}\quad\ldots\quad\ldots\quad \begin{array}{c}\vdots\ \boxed{\Pi_\ell}\\ \Gamma,B_\ell\vdash D\end{array}\quad\ldots}{\Gamma\vdash D}\,\mathrm{el}_r$$

- A_j ranges over all propositions that have a 1 in the truth table of c'; A_i ranges over all propositions that have a 0,
- B_k ranges over all propositions that have a 1 in the truth table of c; B_ℓ ranges over all propositions that have a 0.

The *permutation conversion* is defined by replacing the derivation pattern above by

$$\dfrac{\Gamma\vdash\Psi\ldots\quad \begin{array}{c}\vdots\ \boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\quad\ldots\quad\ldots\quad \dfrac{\begin{array}{c}\vdots\ \boxed{\Sigma_i}\\ \Gamma,A_i\vdash\Phi\end{array}\quad\ldots\quad \begin{array}{c}\vdots\ \boxed{\Pi_k}\\ \Gamma,A_i\vdash B_k\end{array}\quad\ldots\quad\ldots\quad \begin{array}{c}\vdots\ \boxed{\Pi_\ell}\\ \Gamma,A_i,B_\ell\vdash D\end{array}\quad\ldots}{\Gamma,A_i\vdash D}\,\mathrm{el}_r}{\Gamma\vdash D}\,\mathrm{el}_{r'}$$

This gives rise to copying of sub-derivations: for every A_i we copy the sub-derivations $\Pi_1,\ldots,\Pi_n$.

NB. Due to weakening, $\boxed{\Pi_k}$ is also a derivation of $\Gamma, A_i \vdash B_k$ and $\boxed{\Pi_\ell}$ is also a derivation of $\Gamma, A_i, B_\ell \vdash D$.

▶ **Example 26.** If we reduce the permutation convertibility in Example 24, we obtain the following derivation.

$$\dfrac{\Gamma \vdash A \vee B \qquad \dfrac{\dfrac{\dfrac{\Gamma, A, C \vdash C \to D}{\Gamma, A \vdash C \to D}\,\to\text{-in}_a \qquad \Gamma, A \vdash C}{\Gamma, A \vdash D}\,\to\text{-el} \qquad \dfrac{\Gamma, B \vdash C \to D \quad \Gamma, B \vdash C}{\Gamma, B \vdash D}\,\to\text{-el}}{}}{\Gamma \vdash D}\,\vee\text{-el}$$

4 The Curry-Howard isomorphism

We now define typed proof-terms for derivations, which enables the study of "proofs as terms" and emphasis es the computational interpretation of proofs, as detour conversion and permutation conversion will correspond to reductions on these proof-terms. For each connective c we give a general definition of proof-terms for the full set of derivation rules for c, as they have been derived from the truth table. This amounts to a system $\lambda^{\mathcal{C}}$, parametrized by a set of connectives $\mathcal{C}$. Then, to clarify the approach, we show how this works out on a number of examples.

Often, we don't want to consider the full rules for a connective c, but only the optimized rules, following Lemmas 9 and 12. For these optimized rules, there is also a straightforward definition of proof-terms and of the reduction relation associated with (detour, permutation) conversion. In the next Section 5 we show in detail how Lemmas 9 and 12 can be extended to terms and reductions: the proof-terms for the optimized rules can be defined in terms of our original calculus $\lambda^{\mathcal{C}}$, and the reduction rules for the optimized proof terms are an instance of reductions in the original calculus (often multi-step).

▶ **Definition 27.** Given a logic with intuitionistic derivation rules, as derived from truth tables for a set of connectives $\mathcal{C}$, as in Definition 1, we now define the typed λ-calculus $\lambda^{\mathcal{C}}$. The system $\lambda^{\mathcal{C}}$ has judgments $\Gamma \vdash t : A$, where A is a formula, Γ is a set of declarations $\{x_1 : A_1, \ldots, x_m : A_m\}$, where the A_i are formulas and the x_i are term-variables such that every x_i occurs at most once in Γ, and t is a *proof-term.*

Let $c \in \mathcal{C}$ be a connective of arity n, which has 2^n rules (introduction plus elimination rules). For each rule r we have a term: an *introduction term*, $\{\overline{p}\,;\overline{Q}\}_r$, if r is an introduction rule, or an *elimination term*, $t \cdot_r [\overline{p}\,;\overline{Q}]$, if r is an elimination rule. Here, t is again a term, $\overline{p}$ is a finite sequence of terms and $\overline{Q}$ is a finite sequence of *abstracted terms* $\lambda x : A.q$, where x is a term-variable, A is a proposition and q is a term. So the abstract syntax for proof-terms, Term, is as follows.

$$t ::= x \mid \{\overline{t}\,;\overline{\lambda x : A.t}\}_r \mid t \cdot_r [\overline{t}\,;\overline{\lambda x : A.t}]$$

where x ranges over variables and r ranges over the rules of all the connectives.

The terms are *typed* using the following derivation rules.

$$\frac{}{\Gamma \vdash x_i : A_i} \text{ if } x_i : A_i \in \Gamma$$

$$\frac{\ldots \Gamma \vdash p_j : A_j \ldots \quad \ldots \Gamma, y_i : A_i \vdash q_i : \Phi \ldots}{\Gamma \vdash \{\overline{p}\ ; \overline{\lambda y : A.q}\}_r : \Phi} \text{ in}$$

$$\frac{\Gamma \vdash t : \Phi \quad \ldots \Gamma \vdash p_k : A_k \ldots \quad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash t \cdot_r [\overline{p}\ ; \overline{\lambda y : A.q}] : D} \text{ el}$$

Here, $\overline{p}$ is the sequence of terms $p_1, \ldots, p_{m'}$ for all the 1-entries in the truth table, and $\overline{\lambda y : A.q}$ is the sequence of terms $\lambda y_1 : A_1.q_1, \ldots, \lambda y_m : A_m.q_m$ for all the 0-entries in the truth table.

▶ Convention 28. We view the λ-abstracted variables as being *typed* so we write $\overline{\lambda y : A.q}$ and $\lambda y_1 : A_1.q_1, \ldots, \lambda y_m : A_r.q_m$. However, these types clutter up the syntax considerably, so in practice we will almost always leave the types implicit. In case we want to stress that a variable has a certain type, or in case type information enhances the understanding, we will write the type as a superscript, so $\lambda x^A.p$ in stead of $\lambda x : A.p$.

We will sometimes leave the rule r that the elimination or introduction term corresponds to implicit, or we will just number the terms or introduce special names for them without explicit reference to the rule. It should be clear that every line in the truth table for the connective gives rise to one rule, which again gives rise to one term-constructor, which is either an elimination or an introduction term-constructor.

There are term reduction rules that correspond to detour conversion.

▶ **Definition 29.** Given a detour convertibility as defined in Definition 22, we add reduction rules for the associated terms as follows.

- For the $\ell = j$ case, that is, $y_\ell : A_\ell$ and $p_j : A_j$ with $A_\ell = A_j$:

$$\{\overline{p, p_j}\ ; \overline{\lambda x.q}\} \cdot [\overline{s}\ ; \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_a r_\ell[y_\ell := p_j]$$

- For the $k = i$ case, that is, $s_k : A_k$ and $x_i : A_i$ with $A_k = A_i$:

$$\{\overline{p}\ ; \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot [\overline{s, s_k}\ ; \overline{\lambda y.r}] \longrightarrow_a q_i[x_i := s_k] \cdot [\overline{s, s_k}\ ; \overline{\lambda y.r}]$$

For simplicity of presentation we write the "matching cases" in Definition 22 as last term of the sequence. So when writing $\overline{p, p_j}$, this should be understood as a sequence of terms $p_1, \ldots, p_j, \ldots p_{m'}$, where we have singled out the p_j that matches the r_ℓ in $\overline{\lambda y.r, \lambda y_\ell.r_\ell}$. Similarly for $\overline{s, s_k}$ and $\overline{\lambda x.q, \lambda x_i.q_i}$.

It is important to note that there is always (at least one) "matching case", because introduction rules and elimination rules comes from different lines in the truth table.

The reduction is extended in the straightforward way to sub-terms, by defining it as a congruence with respect to the term constructions.

This Definition gives a reduction rule, and possibly more than one, for every combination of an elimination and an introduction. For an n-ary connective, there are 2^n rules in the truth table, and therefore 2^n term-constructors (introduction plus elimination constructors). We now give the examples of the proof-terms for $\vee$ and $\wedge$ in full. In the rules we will always omit the context Γ.

▶ **Example 30.** The rules for disjunction are as follows.

$$\frac{\vdash t : A \vee B \quad x : A \vdash p : D \quad y : B \vdash q : D}{\vdash t \cdot^{\vee} [\,;\lambda x.p, \lambda y.q] : D} \qquad \frac{z : A \vdash r : A \vee B \quad \vdash b : B}{\vdash \{b\,;\lambda z.r\}^{\vee}_{1} : A \vee B}$$

$$\frac{\vdash a : A \quad z : B \vdash r : A \vee B}{\vdash \{a\,;\lambda z.r\}^{\vee}_{2} : A \vee B} \qquad \frac{\vdash a : A \quad \vdash b : B}{\vdash \{a, b\,;\}^{\vee}_{3} : A \vee B}$$

We could have followed our earlier introduced naming convention and index the operators with the line of the truth table they arise from. Then we would write $\{b\,;\lambda z.r\}^{\vee}_{01}$ for $\{b\,;\lambda z.r\}^{\vee}_{1}$, $\{a\,;\lambda z.r\}^{\vee}_{10}$ for $\{a\,;\lambda z.r\}^{\vee}_{2}$ and $\{a, b\,;\}^{\vee}_{11}$ $\{a, b\,;\}^{\vee}_{3}$. This easily clutters up notation, so we don't pursue that.

The reduction rules are

$$\begin{aligned}
\{b\,;\lambda z.r\}^{\vee}_{1} \cdot^{\vee} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a q[y := b]\\
\{a\,;\lambda z.r\}^{\vee}_{2} \cdot^{\vee} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a p[x := a]\\
\{a, b\,;\}^{\vee}_{3} \cdot^{\vee} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a p[x := a]\\
\{a, b\,;\}^{\vee}_{3} \cdot^{\vee} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a q[y := b]
\end{aligned}$$

From the last two cases, we see that the Church-Rosser property (confluence) is lost.

The rules for conjunction are as follows.

$$\frac{\vdash t : A \wedge B \quad x : A \vdash p : D \quad y : B \vdash q : D}{\vdash t \cdot^{\wedge}_{1} [\,;\lambda x.p, \lambda y.q] : D} \qquad \frac{\vdash t : A \wedge B \quad \vdash a : A \quad y : B \vdash q : D}{\vdash t \cdot^{\wedge}_{2} [a\,;\lambda y.q] : D}$$

$$\frac{\vdash t : A \wedge B \quad x : A \vdash p : D \quad \vdash b : B}{\vdash t \cdot^{\wedge}_{3} [b\,;\lambda x.p] : D} \qquad \frac{\vdash a : A \quad \vdash b : B}{\vdash \{a, b\,;\}^{\wedge} : A \wedge B}$$

The reduction rules are

$$\begin{aligned}
\{a, b\,;\}^{\wedge} \cdot^{\wedge}_{1} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a p[x := a]\\
\{a, b\,;\}^{\wedge} \cdot^{\wedge}_{1} [\,;\lambda x.p, \lambda y.q] &\longrightarrow_a q[y := b]\\
\{a, b\,;\}^{\wedge} \cdot^{\wedge}_{2} [a'\,;\lambda y.q] &\longrightarrow_a q[y := b]\\
\{a, b\,;\}^{\wedge} \cdot^{\wedge}_{3} [b'\,;\lambda x.p] &\longrightarrow_a p[x := a]
\end{aligned}$$

From the first two cases, we see that the Church-Rosser property (confluence) is lost.

In Example 39 we will show how we can define proof-terms for the optimized rules for $\wedge$ in terms of the proof-terms for the full rules, while preserving reduction.

In the reduction for the terms for $\vee$ and $\wedge$, an elimination is always removed at each step. The situation gets more interesting with implication.

▶ **Example 31.** The rules for implication are as follows.

$$\frac{x : A \vdash p : A \to B \quad y : B \vdash q : A \to B}{\vdash \{\,;\lambda x.p, \lambda y.q\}^{\to}_{1} : A \to B} \qquad \frac{x : A \vdash p : A \to B \quad \vdash b : B}{\vdash \{b\,;\lambda x.p\}^{\to}_{2} : A \to B}$$

$$\frac{\vdash t : A \to B \quad \vdash a : A \quad z : B \vdash r : D}{\vdash t \cdot^{\to} [a\,;\lambda z.r] : D} \qquad \frac{\vdash a : A \quad \vdash b : B}{\vdash \{a, b\,;\}^{\to}_{3} : A \to B}$$

The reduction rules are

$$\begin{array}{rcl}
\{ \ ; \lambda x.p, \lambda y.q\}^{\to}_1 \cdot^{\to} [a \ ; \lambda z.r] & \longrightarrow_a & p[x := a] \cdot^{\to} [a \ ; \lambda z.r] \\
\{b \ ; \lambda x.p\}^{\to}_2 \cdot^{\to} [a \ ; \lambda z.r] & \longrightarrow_a & r[z := b] \\
\{b \ ; \lambda x.p\}^{\to}_2 \cdot^{\to} [a \ ; \lambda z.r] & \longrightarrow_a & p[x := a] \cdot^{\to} [a \ ; \lambda z.r] \\
\{a', b \ ; \ \}^{\to}_3 \cdot^{\to} [a \ ; \lambda z.r] & \longrightarrow_a & r[z := b]
\end{array}$$

From the second and third case, we can see that Church-Rosser is lost. In the first and the third case, we see that the elimination remains.

In Example 41 we will show how we can define proof-terms for the optimized rules for $\to$ in terms of the proof-terms for the full rules, while preserving reduction. In Definition 48 we will define the standard rules for $\to$.

We now extend the reduction on proof-terms to also capture the permutation conversions of Definition 25. This gives rise to two elimination constructs permuting with each other.

▶ **Definition 32.** Given a permutation convertibility as defined in Definition 25, we add reduction rules for the associated terms as follows.

$$(t \cdot [\overline{p} \ ; \overline{\lambda x.q}]) \cdot [\overline{s} \ ; \overline{\lambda y.r}] \quad \longrightarrow_b \quad t \cdot [\overline{p} \ ; \overline{\lambda x.(q \cdot [\overline{s} \ ; \overline{\lambda y.r}])}]$$

Here, the notation $\overline{\lambda x.(q \cdot [\overline{s} \ ; \overline{\lambda y.r}])}$ should be understood as a sequence $\lambda x_1.q_1, \ldots, \lambda x_m.q_m$ where each q_j is replaced by $q_j \cdot [\overline{s} \ ; \overline{\lambda y.r}]$.

The reduction is extended in the straightforward way to sub-terms, by defining it as a congruence with respect to the term constructions.

▶ Notation 33. We omit brackets by letting the application operator $- \cdot -$ associate to the left, so $t \cdot [\overline{p} \ ; \overline{\lambda x.q}] \cdot [\overline{s} \ ; \overline{\lambda y.r}]$ denotes $(t \cdot [\overline{p} \ ; \overline{\lambda x.q}]) \cdot [\overline{s} \ ; \overline{\lambda y.r}]$. We will also omit the brackets in $\overline{\lambda x.(q \cdot [\overline{s} \ ; \overline{\lambda y.r}])}$, because no ambiguity can arise here.

We treat the well-known example from intuitionistic logic of the $\vee$-elimination, where a permutation convertibility can occur. See also Example 24.

▶ **Example 34.**

$$\frac{\dfrac{\vdash t : A \vee B \quad x : A \vdash p : C \to D \quad y : B \vdash q : C \to D}{\vdash t \cdot^{\vee} [\ ; \lambda x.p, \lambda y.q] : C \to D} \qquad \vdash c : C \quad z : D \vdash r : E}{\vdash t \cdot^{\vee} [\ ; \lambda x.p, \lambda y.q] \cdot^{\to} [c \ ; \lambda z.r] : E}$$

We observe two consecutive elimination rules, where a potential detour convertibility, arising e.g. when q is an introduction term, is blocked by the $\vee$-elimination.

The term reduces as follows

$$t \cdot^{\vee} [\ ; \lambda x.p, \lambda y.q] \cdot^{\to} [c \ ; \lambda z.r] \longrightarrow_b t \cdot^{\vee} [\ ; \lambda x.p \cdot^{\to} [c \ ; \lambda z.r], \lambda y.q \cdot^{\to} [c \ ; \lambda z.r]]$$

We can now easily define the terms in normal-form under the combined reduction $\longrightarrow_{ab}$. The proof is straightforward and comes from the fact that an introduction followed by an elimination is always a redex. (There is always a "matching case" in Definition 29.)

▶ **Lemma 35.** *The set of* terms in normal form *of* $\mathsf{IPC}_{\mathcal{C}}$, NF*is characterized by the following inductive definition.*

- $x \in \mathsf{NF}$ *for every variable* x*,*

- $\{\overline{p}\,;\,\overline{\lambda y.q}\} \in \mathsf{NF}$ *if all* p_i *and* q_j *are in* NF,
- $x \cdot [\overline{p}\,;\,\overline{\lambda y.q}] \in \mathsf{NF}$ *if all* p_i *and* q_j *are in* NF *and* x *is a variable.*

▶ Remark. In [23], yet another notion of convertibility is defined, called *simplification convertibility*. This is a situation where the assumption is unused in an introduction or elimination rule and the rule can be removed all together. Adding these rules is not necessary for the sub-formula property, so we don't introduce it here. On the term level, an elimination of simplification convertibilities would amount to the following reduction rules.

$$\begin{array}{rcll} t \cdot [\overline{p}\,;\,\overline{\lambda x.q}] & \longrightarrow & q_i & \text{if } x_i \notin \mathrm{FV}(q_i) \\ \{\overline{p}\,;\,\overline{\lambda x.q}\} & \longrightarrow & q_i & \text{if } x_i \notin \mathrm{FV}(q_i) \end{array}$$

5 Extending the Curry-Howard isomorphism to definable rules

The optimizations for the logical rules, as given in Lemmas 9 and 12 can be extended to the proof terms and also to convertibilities and conversions. This gives us the possibility to capture questions related to normalization by looking at normalization for terms in the original calculus $\lambda^{\mathcal{C}}$. We will now describe the terms for the optimized rules in detail.

▶ **Definition 36.** For each optimization step in Lemmas 9 and 12 we give the canonical term for the optimized rule and its translation in terms of $\lambda^{\mathcal{C}}$ of Definition 27.

We first treat the two optimizations arising from Lemma 9, and then the optimization arising from Lemma 12.

- Given two rules

$$\frac{\vdash p_1 : A_1 \;\ldots\; \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : \Phi \;\ldots\; x_m : B_m \vdash q_m : \Phi \qquad z : A \vdash s : \Phi}{\vdash \{\overline{p}\,;\,\overline{\lambda x.q, \lambda z.s}\}_r : \Phi}\,\mathrm{in}_r$$

$$\frac{\vdash p_1 : A_1 \;\ldots\; \vdash p_n : A_n \qquad \vdash a : A \qquad x_1 : B_1 \vdash q_1 : \Phi \;\ldots\; x_m : B_m \vdash q_m : \Phi}{\vdash \{\overline{p,a}\,;\,\overline{\lambda x.q}\}_{r'} : \Phi}\,\mathrm{in}_{r'}$$

we have the following term for the optimized introduction rule

$$\frac{\vdash p_1 : A_1 \;\ldots\; \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : \Phi \;\ldots\; x_m : B_m \vdash q_m : \Phi}{\vdash \{\overline{p}\,;\,\overline{\lambda x.q, \lambda z.\{\overline{p,z}\,;\,\overline{\lambda x.q}\}_{r'}}\}_r : \Phi}\,\mathrm{in}^{\mathrm{opt}}_{r,r'}$$

We define the term $\{\overline{p}\,;\,\overline{\lambda x.q}\}^{\circ}_{r,r'}$ as $\{\overline{p}\,;\,\overline{\lambda x.q, \lambda z.\{\overline{p,z}\,;\,\overline{\lambda x.q}\}_{r'}}\}_r$

- Given two rules

$$\frac{\vdash t : \Phi \quad \vdash p_1 : A_1 \ldots \vdash p_n : A_n \quad x_1 : B_1 \vdash q_1 : D \ldots x_m : B_m \vdash q_m : D \quad z : A \vdash s : D}{\vdash t \cdot_r [\overline{p}\,;\,\overline{\lambda x.q, \lambda z.s}] : D}\,\mathrm{el}_r$$

$$\frac{\vdash t : \Phi \quad \vdash p_1 : A_1 \ldots \vdash p_n : A_n \quad \vdash a : A \quad x_1 : B_1 \vdash q_1 : D \ldots x_m : B_m \vdash q_m : D}{\vdash t \cdot_{r'} [\overline{p,a}\,;\,\overline{\lambda x.q}] : D}\,\mathrm{el}_{r'}$$

we have the following term for the optimized elimination rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \;\ldots\; \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : D \;\ldots\; x_m : B_m \vdash q_m : D}{\vdash t \cdot_r [\overline{p}\,;\,\overline{\lambda x.q, \lambda z.t \cdot_{r'} [\overline{p,z}\,;\,\overline{\lambda x.q}]}] : D}\,\mathrm{el}^{\mathrm{opt}}_{r,r'}$$

We define term $t \odot_{r,r'} [\overline{p}\,;\,\overline{\lambda x.q}]$ as $t \cdot_r [\overline{p}\,;\,\overline{\lambda x.q, \lambda z.t \cdot_{r'} [\overline{p,z}\,;\,\overline{\lambda x.q}]}]$

- Given the rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad z : A \vdash s : D}{\vdash t \cdot_r [\overline{p} \ ; \ \lambda z.s] : D}\ \mathrm{el}_r$$

we have the following term for the optimized elimination rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n}{\vdash t \cdot_r [\overline{p} \ ; \ \lambda z.z] : A}\ \mathrm{el}_r^{\mathrm{opt}}$$

We define the term $t \boxdot_r [\overline{p}]$ as $t \cdot_r [\overline{p} \ ; \ \lambda z.z]$

There is a canonical way in which the notions of detour convertibility and detour conversion extend to the optimized rules: the same rules as in Definition 29 apply. In case of a term of the form $\{\dots \ ; \ \dots\} \cdot [\dots \ ; \ \dots]$, a reduction is always possible, also in the case of optimized rules. For the permutation convertibilities, the situation is similar: the same rules as in Definition 32 apply.

▶ **Definition 37.** We define reduction on the optimized terms as follows. Let $\oslash$ be any $\cdot_{r''}$ or $\odot_{r'',r'''}$ for some r'', r'''. (For the notation, we refer to Definition 29.)

$$\begin{array}{rcl}
\text{For the } \ell = j \text{ case:} & & \\
\{\overline{p, p_j} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \oslash [\overline{s} \ ; \ \overline{\lambda y.u, \lambda y_\ell.u_\ell}] & \longrightarrow_a & u_\ell[y_\ell := p_j] \\
\text{For the } k = i \text{ case:} & & \\
\{\overline{p} \ ; \ \overline{\lambda x.q, \lambda x_i.q_i}\}^{\circ}_{r,r'} \oslash [\overline{s, s_k} \ ; \ \overline{\lambda y.u}] & \longrightarrow_a & q_i[x_i := s_k] \oslash [\overline{s, s_k} \ ; \ \overline{\lambda y.u}] \\
\text{For the } k = i \text{ case:} & & \\
\{\overline{s} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \boxdot_r [\overline{p}] & \longrightarrow_a & q_i[x_i := p_k] \boxdot_r [\overline{p}] \\
\text{Special case:} & & \\
\{\overline{s, s_j} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \boxdot_r [\overline{p}] & \longrightarrow_a & s_j
\end{array}$$

The last special case is when $\{\overline{s, s_j} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \boxdot_r [\overline{p}] : A$ and $s_j : A$. See the definition of $\{\overline{s, s_j} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \boxdot_r [\overline{p}]$ as $\{\overline{s, s_j} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'} \cdot_r [\overline{p} \ ; \ \lambda z.z]$ in Definition 36; this is the case where s_j matches the "invisible" $\lambda z.z$.

We also extend the notions of permutation convertibility and permutation conversion from Definition 25 (see also Definition 32): we add reduction rules for the optimized terms as follows.

$$(t \ominus [\overline{p} \ ; \ \overline{\lambda x.q}]) \oslash [\overline{s} \ ; \ \overline{\lambda y.u}] \quad \longrightarrow_b \quad t \ominus [\overline{p} \ ; \ \overline{\lambda x.(q \oslash [\overline{s} \ ; \ \overline{\lambda y.u}])}]$$

where $\ominus$ is any $\cdot_{r''}$ or $\odot_{r'',r'''}$ and $\oslash$ is any $\cdot_{r''}$ or $\odot_{r'',r'''}$ or $\boxdot_{r''}$.

▶ Remark. To clarify, we want to note explicitly that $t \boxdot_r [\overline{p}] \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}]$ does *not reduce to* $t \boxdot_r [\overline{p}]$. In case we only have the optimized rules, it does not reduce at all. If we consider $t \boxdot_r [\overline{p}]$ as a definition in the original calculus $\lambda^{\mathcal{C}}$, we do have a reduction,

$$t \boxdot_r [\overline{p}] \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}] \longrightarrow_b t \cdot_r [\overline{p} \ ; \ \lambda z.z \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}]]$$

but this uses a non-optimized elimination.

▶ Remark. The process described in Definition 36, which is based on Lemmas 9 and 12 can be iterated, as we have seen in earlier examples. A simple way to view the rules for an n-ary connective c as a pair (b, r) where b is 0 or 1 and r is a partial function $r : \{1, 2, \dots, n\} \to \{0, 1\}$. For a standard rule, derived from a line in the the truth table of c,

r is a total function. (If $r(i) = 1$, then A_i is a lemma in the rule and if $r(j) = 0$, then A_j is a case; if $b = 0$, we have an elimination rule, if $b = 1$ we have an introduction rule .) An optimized rule is a function r that is undefined for some elements of $\{1, \ldots, n\}$.

For the first case of Definition 36, where $\{\ldots\,;\,\ldots\}^{\circ}_{r,r'}$ is defined in terms of $\{\ldots\,;\,\ldots\}_r$ and $\{\ldots\,;\,\ldots\}_{r'}$, we have $r'' = r \cap r'$ for the optimized rule r''. This is allowed in case $b = 1$ for r and r' and r and r' differ for only one element.

For the second case of Definition 36, where $\ldots \odot_{r,r'} [\ldots\,;\,\ldots]$ is defined in terms of $\ldots \cdot_r [\ldots\,;\,\ldots]$ and $\ldots \cdot_{r'} [\ldots\,;\,\ldots]$, we again have $r'' = r \cap r'$ for the optimized rule r''. This is allowed in case $b = 0$ for r and r' and r and r' differ for only one element.

Optimization according to Lemma 12, the third case of Definition 36, corresponds with a (possibly partial) function r where $b = 0$ and $r(i) = 1$ for exactly one i.

With the definable optimized terms for elimination and introduction, we have a choice of taking these as defined terms, or taking them as primitives and removing the originals. Or even there is a third alternative of adding them as additional term constructions. After we have done some examples, we will, in Lemma 43, analyze the reduction behaviour of the newly defined terms in terms of the original ones.

Before that we state what the normal forms are of the optimized terms and the optimized reduction, extending Lemma 35. So in the following Lemma, we consider the situation where we have added optimized terms and reductions, while removing the original ones. The proof is straightforward, keeping in mind Remark 5 and the fact that with optimized terms, if an introduction is followed immediately by an elimination, then there is a "matching case" that allows us to reduce the term.

▶ **Lemma 38.** *We simultaneously characterize* $\mathsf{NF}^{\mathrm{opt}}$*, the set of* terms in normal form *of* $\mathsf{IPC}_{\mathcal{C}}$ *with optimized terms and reductions, and the set of* neutral terms *inductively as follows.*

- $x \in \mathsf{NF}^{\mathrm{opt}}$ *and x is neutral, for every variable x,*
- $\{\overline{p}\,;\,\overline{\lambda y.q}\} \in \mathsf{NF}^{\mathrm{opt}}$ *if all p_i and q_j are in* $\mathsf{NF}^{\mathrm{opt}}$*,*
- $x \oslash [\overline{p}\,;\,\overline{\lambda y.q}] \in \mathsf{NF}^{\mathrm{opt}}$ *if all p_i and q_j are in* $\mathsf{NF}^{\mathrm{opt}}$ *and x is a variable; this term is neutral if $\oslash = \boxdot_r$ for some r.*
- $t \boxdot_r [\overline{s}] \oslash [\overline{p}\,;\,\overline{\lambda y.q}] \in \mathsf{NF}^{\mathrm{opt}}$ *if all s_k, p_i and q_j are in* $\mathsf{NF}^{\mathrm{opt}}$ *and t is neutral; this term is neutral if $\oslash = \boxdot_{r'}$ for some r'.*

What the Lemma says is that terms like

$$x \boxdot_r [\overline{s_1}] \boxdot_{r'} [\overline{s_2}] \boxdot_{r''} \ldots\ldots \oslash [\overline{p}\,;\,\overline{\lambda y.q}]$$

are also normal forms, if $\overline{s_1}$, $\overline{s_2}, \ldots, \overline{p}$ and $\overline{q}$ are.

▶ **Example 39.** We continue Example 30 and look into the optimized rules for $\wedge$, as given in Definition 13. The introduction rule of Example 30 is the same as in Definition 13; the usual "pairing" construction is given by $\{a, b\,;\,\}^{\wedge}$. For elimination, we would like to have the following "projection" rules.

$$\frac{\vdash t : A \wedge B}{\vdash \pi_1\, t : A} \qquad \frac{\vdash t : A \wedge B}{\vdash \pi_2\, t : B}$$

That is, we would like to define $\pi_1\, t$ and $\pi_2\, t$ in terms of the constructions of Example 30, with the expected reduction rules: $\pi_1\, \{a, b\,;\,\}^{\wedge} \longrightarrow_a a$ and $\pi_2\, \{a, b\,;\,\}^{\wedge} \longrightarrow_a b$. Definition 36 gives the clue. Let's consider the first projection, $\pi_1\, t$. We have the following optimization of the $\wedge$-rules of Example 30.

$$\frac{\vdash t : A \wedge B \qquad x : A \vdash p : D}{t \odot^{\wedge}_{a} [\,;\, \lambda x^A.p] : D}$$

where $t \odot_a^{\wedge} [\,;\, \lambda x^A.p] := t \cdot_1^{\wedge} [\,;\, \lambda x^A.p, \lambda z^B.t \cdot_3^{\wedge} [z\,;\, \lambda x^A.p]]$. It is easily verified that we have the following reduction

$$\{a, b\,;\,\}^{\wedge} \odot_a^{\wedge} [\,;\, \lambda x^A.p] \longrightarrow_a p[x := a].$$

We have another optimization:

$$\frac{\vdash t : A \wedge B}{\vdash t \boxdot_1^{\wedge} [\,;\,] : A}$$

where $t \boxdot_1^{\wedge} [\,;\,] := t \odot_a^{\wedge} [\,;\, \lambda x^A.x]$.

All together we have

$$\pi_1\, t := t \boxdot_1^{\wedge} [\,;\,] = t \odot_a^{\wedge} [\,;\, \lambda x^A.x] = t \cdot_1^{\wedge} [\,;\, \lambda x^A.x, \lambda z^B.t \cdot_3^{\wedge} [z\,;\, \lambda x^A.x]]$$

which has the following reductions.

$$\begin{array}{rcl}
\pi_1 \{a, b\,;\,\}^{\wedge} & = & \{a, b\,;\,\}^{\wedge} \cdot_1^{\wedge} [\,;\, \lambda x^A.x, \lambda z^B.\{a, b\,;\,\}^{\wedge} \cdot_3^{\wedge} [z\,;\, \lambda x^A.x]] \\
& \longrightarrow_a & a \\
\pi_1 \{a, b\,;\,\}^{\wedge} & = & \{a, b\,;\,\}^{\wedge} \cdot_1^{\wedge} [\,;\, \lambda x^A.x, \lambda z^B.\{a, b\,;\,\}^{\wedge} \cdot_3^{\wedge} [z\,;\, \lambda x^A.x]] \\
& \longrightarrow_a & \{a, b\,;\,\}^{\wedge} \cdot_3^{\wedge} [b\,;\, \lambda x^A.x] \\
& \longrightarrow_a & a
\end{array}$$

Similarly, we define $\pi_2\, t := t \cdot_1^{\wedge} [\,;\, \lambda x^B.x, \lambda z^A.t \cdot_2^{\wedge} [z\,;\, \lambda x^B.x]]$. Then $\pi_2 \{a, b\,;\,\}^{\wedge} \longrightarrow_a^{+} b$.

An interesting feature is that the reduction rules for our non-optimized calculus are not Church-Rosser, as we have already indicated in Example 30 and also in Example 23. On the other hand, the optimized rules for standard intuitionistic proposition logic are know to be Church-Rosser. We look into the case for $\wedge$ in more detail.

▶ **Example 40.** The set of full rules for $\wedge$, see Example 30, is not Church-Rosser as the following concrete example shows. Suppose we have $\vdash p : D$ and $\vdash q : D$, where p and q are different.

$$\cfrac{\cfrac{a : A \vdash a : A \qquad b : B \vdash b : B}{a : A, b : B \vdash \{a, b\,;\,\}^{\wedge} : A \wedge B} \qquad x : A \vdash p : D \qquad y : B \vdash q : D}{\{a, b\,;\,\}^{\wedge} \cdot_1^{\wedge} [\,;\, \lambda x^A.p, \lambda y^B, q]}$$

This term reduces to both p and q, which are distinct terms of type D. The crucial point is in the rule for $- \cdot_1^{\wedge} [\,;\, -]$ that admits a choice:

$$\frac{\vdash t : A \wedge B \quad x : A \vdash p : D \quad y : B \vdash q : D}{\vdash t \cdot_1^{\wedge} [\,;\, \lambda x.p, \lambda y.q] : D}$$

For $t = \{a, b\,;\,\}^{\wedge}$ we can either select the "A-case" or the "B-case".

We have shown how the optimized rules can be explained in terms of the full rules, but we can also doe the opposite: interpret the full rules for $\wedge$ of Example 30 in terms of π_1 and π_2. Then we get

$$\begin{array}{rcl}
t \cdot_1^{\wedge} [\,;\, \lambda x^A.p, \lambda y^B.q] & := & p[x := \pi_1\, t] \\
t \cdot_2^{\wedge} [a'\,;\, \lambda y^B.q] & := & q[y := \pi_2\, t] \\
t \cdot_3^{\wedge} [b'\,;\, \lambda x^A.p] & := & p[x := \pi_1\, t]
\end{array}$$

where in the first case we could also have chosen $q[y := \pi_2\, t]$. We observe that the non-determinism in the full rules is resolved by a choice we make in the translation of the first $\wedge$-elimination.

▶ **Example 41.** We now look into the optimized rules for implication of Definition 13. The full rules have been treated in Example 31. We want to define the following terms.

$$\frac{x : A \vdash p : A \to B}{\vdash \{\,;\,\lambda x^A.p\}_1^{\to\circ} : A \to B} \qquad \frac{\vdash b : B}{\vdash \{b\,;\,\}_2^{\to\circ} : A \to B} \qquad \frac{\vdash t : A \to B \quad \vdash a : A}{\vdash t \boxdot^{\to} [a] : B}$$

These can be defined from the terms in Example 31 via the optimizations of Definition 36 as follows.

$$\begin{array}{rcl} \{\,;\,\lambda x^A.p\}_1^{\to\circ} & := & \{\,;\,\lambda x^A.p, \lambda z.\{z\,;\,\lambda x^A.p\}_2^{\to}\}_1^{\to} \\ \{b\,;\,\}_2^{\to\circ} & := & \{b\,;\,\lambda z^A.\{z, b\,;\,\}_3^{\to}\}_2^{\to} \\ t \boxdot^{\to} [a] & := & t \bullet^{\to} [a\,;\,\lambda z.z] \end{array}$$

These obey the following reductions.

$$\begin{array}{rcl} \{\,;\,\lambda x^A.p\}_1^{\to\circ} \boxdot^{\to} [a] & = & \{\,;\,\lambda x^A.p, \lambda z.\{z\,;\,\lambda x^A.p\}_2^{\to}\}_1^{\to} \bullet^{\to} [a\,;\,\lambda z.z] \\ & \longrightarrow_a & p[x := a] \bullet^{\to} [a\,;\,\lambda z.z] \\ & = & p[x := a] \boxdot^{\to} [a] \\ \{b\,;\,\}_2^{\to\circ} \boxdot^{\to} [a] & := & \{b\,;\,\lambda z^A.\{z, b\,;\,\}_3^{\to}\}_2^{\to} \boxdot^{\to} [a] \\ & \longrightarrow_a & b \\ \{b\,;\,\}_2^{\to\circ} \boxdot^{\to} [a] & := & \{b\,;\,\lambda z^A.\{z, b\,;\,\}_3^{\to}\}_2^{\to} \boxdot^{\to} [a] \\ & \longrightarrow_a & \{a, b\,;\,\}_3^{\to} \boxdot^{\to} [a] \\ & \longrightarrow_a & b \end{array}$$

These are the exact reduction rules one would expect for these terms. We can again translate these to the well-known β-rules, that we will define in Definition 47.

The definition of the standard rule for $\to$-introduction essentially uses the $\boxdot$ construction, which has a somewhat special behaviour under normalization, as we have seen in Remark 5 and Lemma 38. Let's look at an example to emphasis this.

▶ **Example 42.** Consider the following proof.

$$\frac{\dfrac{t : A \to B \to C \qquad \vdash a : A}{t \boxdot^{\to} [a] : B \to C} \qquad \vdash b : B}{t \boxdot^{\to} [a] \boxdot^{\to} [b] : C}$$

If t is not an introduction term ($t \neq \{\lambda x.q\}^{\to}$), then this is not a redex with the optimized rules. However, in case $\boxdot$ is a defined term-construction, this term is reducible:

$$t \boxdot^{\to} [a] \boxdot^{\to} [b] \longrightarrow_b t \bullet^{\to} [a\,;\,\lambda z^{B\to C}.z \boxdot^{\to} [b]].$$

To clarify, the derivation for this term is:

$$\frac{\vdash t : A \to B \to C \qquad \vdash a : A \qquad \dfrac{z : B \to C \vdash z : B \to C \qquad \vdash b : B}{z : B \to C \vdash z \boxdot^{\to} [b] : C}}{t \bullet^{\to} [a\,;\,\lambda z^{B\to C}.z \boxdot^{\to} [b]] : C}$$

▶ **Lemma 43.** *The translation of an $\longrightarrow_a$ step in the optimized calculus translates to a (possibly multistep) $\longrightarrow_a$ step in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. We show two cases:

1. If $\{\overline{t} \,;\, \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p} \,;\, \overline{\lambda x.q}] \longrightarrow_a R$ (using the optimized rules) and $\{\overline{t} \,;\, \overline{\lambda y.r}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p} \,;\, \overline{\lambda x.q}]$ translates to T in the original calculus $\lambda^{\mathcal{C}}$, then there is a term T' such that $T \longrightarrow_a^+ T'$ and R translates to T' in $\lambda^{\mathcal{C}}$. Here $\longrightarrow_a^+$ denotes a non-zero sequence of reductions.

   In this case the translation T is as follows. $T = M \cdot [\overline{p} \,;\, \overline{\lambda x.q, \lambda z.\overline{M \cdot [\overline{p,z} \,;\, \overline{\lambda x.q}]}}]$, where we abbreviate $M := \{\overline{t} \,;\, \overline{\lambda y.v, \lambda z.\{\overline{t,z} \,;\, \overline{\lambda y.v}\}}\}$. There are two possible cases for the reduction.

   - Case $\{\overline{t} \,;\, \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p} \,;\, \overline{\lambda x.q}] \longrightarrow_a q_\ell[x_\ell := t_j]$. Then $T \longrightarrow_a q_\ell[x_\ell := t_j]$ and we are done.
   - Case $\{\overline{t} \,;\, \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p} \,;\, \overline{\lambda x.q}] \longrightarrow_a v_i[y_i := p_k] \odot_{r_1,r_2} [\overline{p} \,;\, \overline{\lambda x.q}]$. Then

   $$\begin{aligned} T &\longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p} \,;\, \overline{\lambda x.q, \lambda z.M \cdot [\overline{p,z} \,;\, \overline{\lambda x.q}]}] \\ &\longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p} \,;\, \overline{\lambda x.q, \lambda z.v_i[y_i := p_k] \cdot [\overline{p,z} \,;\, \overline{\lambda x.q}]}] \end{aligned}$$

     and we are done.

2. If $\{\overline{t} \,;\, \overline{\lambda y.v}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}] \longrightarrow_a R$ and $\{\overline{t} \,;\, \overline{\lambda y.r}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}]$ translates to T in the original calculus $\lambda^{\mathcal{C}}$, then there is a term T' such that $T \longrightarrow_a^+ T'$ and R translates to T' in $\lambda^{\mathcal{C}}$. Now the translation T is as follows. $T = \{\overline{t} \,;\, \overline{\lambda y.v, \lambda z.\{\overline{t,z} \,;\, \overline{\lambda y.v}\}}\} \cdot [\overline{p} \,;\, \lambda z.z]$. There is one possibility for the reduction.

   - Case $\{\overline{t} \,;\, \overline{\lambda y.v}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}] \longrightarrow_a v_i[y_i := p_k] \odot_{r_1} [\overline{p}]$. Then

   $$T \longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p} \,;\, \lambda z.z]$$

     and we are done.

◀

As mentioned, Schroeder-Heister[17] has proposed another elimination rule for $\wedge$ which is slightly different from ours. Von Plato [23] calls this *general elimination* while Tennant [21] calls it *parallel elimination.* We call it parallel $\wedge$-elimination and give it in typed λ-calculus format.

▶ **Definition 44.** We define the *parallel* $\wedge$*-elimination rule* as follows

$$\frac{\Gamma \vdash t : A \wedge B \qquad \Gamma, x : A, y : B \vdash q : D}{\Gamma \vdash t \cdot^{\mathrm{par}} [\lambda x, y.q] : D}\ \wedge\text{-el}$$

The reduction (detour conversion) rule associated with this rule is as follows.

$$\{a, b \,;\, \} \cdot^{\mathrm{par}} [\lambda x, y.q] \longrightarrow_{\mathrm{par}} q[x := a, y := b].$$

We show that this elimination rule can be translated in terms of ours and that reduction is preserved.

▶ **Definition 45.** We translate the parallel $\wedge$-elimination rule of Definition 44 by defining it in terms of the optimized terms for $\wedge$ of Example 39. We consider the following optimized rules, the first of which was given explicitly in Example 39.

$$\frac{\Gamma \vdash t : A \wedge B \qquad \Gamma, x : A \vdash q : D}{\Gamma \vdash t \odot^{\wedge}_{a} [\,;\, \lambda x.q] : D} \qquad \frac{\Gamma \vdash t : A \wedge B \qquad \Gamma, y : B \vdash q : D}{\Gamma \vdash t \odot^{\wedge}_{b} [\,;\, \lambda y.q] : D}$$

Now define

$$t \cdot^{\mathrm{par}} [\lambda x, y.q] := t \odot^{\wedge}_{1} [\,;\, \lambda x.t \odot^{\wedge}_{2} [\,;\, \lambda y.q]].$$

▶ **Lemma 46.** *The defined term $t \cdot^{\mathrm{par}} [\lambda x, y.q]$ is of the right type and the translation of an $\longrightarrow_{\mathrm{par}}$ step in the calculus with the parallel $\wedge$-elimination rule translates to multistep $\longrightarrow_a$ in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. Given $\Gamma \vdash t : A \wedge B$ and $\Gamma, x : A, y : B \vdash q : D$, we have

$$\dfrac{\Gamma \vdash t : A \wedge B \qquad \dfrac{\Gamma, x : A \vdash t : A \wedge B \qquad \Gamma, x : A, y : B \vdash q : D}{\Gamma, x : A \vdash t \odot_2^{\wedge} [\,;\, \lambda y.q] : D}}{\Gamma \vdash t \odot_1^{\wedge} [\,;\, \lambda x.t \odot_2^{\wedge} [\,;\, \lambda y.q]] : D}$$

The reduction can easily be verified:

$$\begin{aligned}
\{a, b\,;\,\}^{\wedge} \cdot^{\mathrm{par}} [\lambda x, y.q] \quad &:= \quad \{a, b\,;\,\}^{\wedge} \odot_1^{\wedge} [\,;\, \lambda x.\{a, b\,;\,\}^{\wedge} \odot_2^{\wedge} [\,;\, \lambda y.q]] \\
&\longrightarrow_a \quad \{a, b\,;\,\}^{\wedge} \odot_2^{\wedge} [\,;\, \lambda y.q[x := a]] \\
&\longrightarrow_a \quad q[x := a, y := b].
\end{aligned}$$

◀

We define the standard rule for $\to$-introduction and show that this introduction rule can be translated in terms of ours and that the reduction is preserved.

▶ **Definition 47.** We define the *standard rule for $\to$-introduction* as follows, where we describe it using terms.

$$\dfrac{\Gamma, x : A \vdash q : B}{\Gamma \vdash \{\lambda x.q\}^{\to} : A \to B} \to\text{-in}$$

The reduction rule associated with this term is as follows.

$$\{\lambda x.q\}^{\to} \boxdot^{\to} [a] \longrightarrow_s q[x := a],$$

where $t \boxdot^{\to} [a]$ is the optimized elimination rule from Example 41.

▶ **Definition 48.** We define the standard $\to$-introduction rule in terms of optimized $\to$-rules (Example 41) as follows. Given $\Gamma, x : A \vdash q : B$ we define

$$\{\lambda x.q\}^{\to} := \{\,;\, \lambda x.\{q\,;\,\}_2^{\to\circ}\}_1^{\to\circ}.$$

▶ **Lemma 49.** *The translation of $\{\lambda x.q\}^{\to}$ is well-typed and the translation of an $\longrightarrow_s$ step in the calculus with the standard rule for $\to$ translates to multistep $\longrightarrow_a$ in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. The well-typedness is easily verified:

$$\dfrac{\dfrac{x : A \vdash q : B}{x : A \vdash \{q\,;\,\}_2^{\to\circ} : A \to B}}{\vdash \{\,;\, \lambda x^A.\{q\,;\,\}_2^{\to\circ}\}_1^{\to\circ} : A \to B}$$

For the reduction:

$$\{\,;\, \lambda x^A.\{q\,;\,\}_2^{\to\circ}\}_1^{\to\circ} \cdot^{\to} [a\,;\,] \longrightarrow_a \{q[x := a]\,;\,\}_2^{\to\circ} \cdot^{\to} [a\,;\,] \longrightarrow_a q[x := a].$$

◀

We define the traditional rule for $\neg$-introduction and show that it can be translated in terms of ours and that detour conversion is preserved.

▶ **Definition 50.** We define the *traditional rules for* ¬, the introduction and the elimination rule, as follows, where we describe them using terms.

$$\frac{\Gamma, x : A \vdash t : \neg B \qquad \Gamma, y : A \vdash q : B}{\Gamma \vdash \{\lambda x.t, \lambda y.q\}^{\mathrm{t}} : \neg A} \qquad \frac{\Gamma \vdash t : \neg A \qquad \Gamma \vdash a : A}{\Gamma \vdash t \cdot^{\neg} [a\ ;\ ] : D}$$

The reduction rule associated with these terms is as follows.

$$\{\lambda x^A.t, \lambda y^A.q\}^{\mathrm{t}} \cdot^{\neg} [a\ ;\ ] \longrightarrow_{\neg} t[x := a] \cdot^{\neg} [q[y := a]\ ;\ ].$$

▶ **Example 51.** The rules for negation that we derive from our general Definition 27 are the following.

$$\frac{\Gamma, x : A \vdash q : \neg A}{\Gamma \vdash \{\ ;\ \lambda x.q\}^{\neg} : \neg A} \qquad \frac{\Gamma \vdash t : \neg A \qquad \Gamma \vdash a : A}{\Gamma \vdash t \cdot^{\neg} [a\ ;\ ] : D}$$

with reduction

$$\{\ ;\ \lambda x^A.q\}^{\neg} \cdot^{\neg} [a\ ;\ ] \longrightarrow_a q[x := a] \cdot^{\neg} [a\ ;\ ].$$

We see that the elimination rule for ¬ in Example 51 is the same as the traditional one. The traditional introduction rule for ¬ is definable.

▶ **Definition 52.** We define the traditional ¬-introduction rule in terms of the one of Example 51 as follows. Given $\Gamma, x : A \vdash t : \neg B$ and $\Gamma, y : A \vdash q : B$ we define

$$\{\lambda x^A.t, \lambda y^A.q\}^{\mathrm{t}} := \{\ ;\ \lambda x^A.t \cdot^{\neg} [q[y := x]\ ;\ ]\}^{\neg}$$

▶ **Lemma 53.** *The definition of* $\{\lambda x.t, \lambda y.q\}^{\mathrm{t}}$ *is well-typed and a* $\longrightarrow_{\neg}$ *step in the calculus with the traditional rule for* ¬ *translates to multistep* $\longrightarrow_a$ *in the original calculus* $\lambda^{\mathcal{C}}$*.*

Proof. For the well-typedness:

$$\cfrac{\cfrac{\Gamma, x : A \vdash t : \neg B \qquad \cfrac{\Gamma, y : A \vdash q : B}{\Gamma, x : A \vdash q[y := x] : B}}{\Gamma, x : A \vdash t \cdot^{\neg} [q[y := x]\ ;\ ] : \neg A}\ \neg\text{-el}}{\Gamma \vdash \{\ ;\ \lambda x^A.t \cdot^{\neg} [q[y := x]\ ;\ ]\}^{\neg} : \neg A}\ \neg\text{-in}$$

For the reduction:

$$\{\ ;\ \lambda x^A.t \cdot^{\neg} [q[y := x]\ ;\ ]\}^{\neg} \cdot^{\neg} [a\ ;\ ] \longrightarrow_a t[x := a] \cdot^{\neg} [q[x := a]\ ;\ ].$$

◀

As a final example, we give the proof-terms for the optimized rules of nand-logic, as described in Definition 14.

▶ **Example 54.** The proof-terms for nand-logic are

$$\frac{x : A \vdash p : A \uparrow B}{\vdash \{\ ;\ \lambda x^A.p\}^{\uparrow} : A \uparrow B} \qquad \frac{y : B \vdash q : A \uparrow B}{\vdash \{\ ;\ \lambda y^B.q\}^{\uparrow} : A \uparrow B} \qquad \frac{\vdash t : A \uparrow B \quad \vdash a : A \quad \vdash b : B}{\vdash t \cdot^{\uparrow} [a, b\ ;\ ] : D}$$

with reduction rules

$$\begin{aligned} \{\ ;\ \lambda x^A.p\}^{\uparrow} \cdot^{\uparrow} [a, b\ ;\ ] &\longrightarrow_a p[x := a] \cdot^{\uparrow} [a, b\ ;\ ] \\ \{\ ;\ \lambda y^B.q\}^{\uparrow} \cdot^{\uparrow} [a, b\ ;\ ] &\longrightarrow_a q[y := b] \cdot^{\uparrow} [a, b\ ;\ ] \end{aligned}$$

This time we have a situation where a permutation conversion actually reduces the size of a term considerably. Suppose $t : A \uparrow B$ and $a : A$, $b : B$, $c : C$, $d : D$. Then we have

$$\dfrac{\dfrac{\vdash t : A \uparrow B \quad \vdash a : A \quad \vdash b : B}{\vdash t \cdot^{\uparrow} [a, b\ ;\ ] : C \uparrow D} \qquad \vdash c : C \qquad \vdash d : D}{t \cdot^{\uparrow} [a, b\ ;\ ] \cdot^{\uparrow} [c, d\ ;\ ] : E}$$

We have

$$t \cdot^{\uparrow} [a, b\ ;\ ] \cdot^{\uparrow} [c, d\ ;\ ] \longrightarrow_b t \cdot^{\uparrow} [a, b\ ;\ ]$$

which is of type E, and we see that the superfluous second nand-elimination rule has been removed.

As another example, we can give a proof-term of $A \vee \neg A^{\uparrow}$, the proposition in nand-logic that we have shown to be provable after the proof of Proposition 17. It's proof-term is

$$\{\ ;\ \lambda x.\{\ ;\ \lambda y.y \cdot^{\uparrow} [x, x\ ;\ ]\}^{\uparrow}\}^{\uparrow} : (A \uparrow A) \uparrow (\dot{\neg} A \uparrow \dot{\neg} A)$$

6 Normalization

In this section we prove that $\longrightarrow_a$ and $\longrightarrow_b$ are both strongly normalizing (SN). We also give a proof of weak normalization (WN) of the combination of $\longrightarrow_a$ and $\longrightarrow_b$. As usual, SN states that there are no terms that have an infinite reduction path, and WN states that for each term there is a reduction path that leads to a normal form. For the proof of WN we describe an actual procedure for finding a normal form of a term.

▶ **Theorem 55.** *The reduction $\longrightarrow_b$ is strongly normalizing.*

Proof. We define a measure $|-|$ from terms to natural numbers that decreases with every reduction step. For notational convenience we suppress the reference to the derivation rule r.

$$\begin{array}{rcl} |x| & := & 1 \\ |\{\overline{p}\ ;\ \overline{\lambda y.q}\}| & := & \Sigma|p_i| + \Sigma|q_j| \\ |t \cdot [\overline{s}\ ;\ \overline{\lambda y.u}]| & := & |t|(2 + \Sigma|s_k| + \Sigma|u_\ell|) \end{array}$$

It can easy be verified that, if $t_0 \longrightarrow_b t_1$, then $|t_0| > |t_1|$, so $\longrightarrow_b$ is strongly normalizing. ◀

▶ **Corollary 56.** *The reduction $\longrightarrow_b$ for the optimized rules of Definition 36, the standard rule for $\to$-elimination of Definition 47, the parallel $\wedge$-elimination rule of Definition 44 and the traditional rule for $\neg$-elimination of Definition 50 are strongly normalizing.*

Proof. The same metrics as in the proof of Theorem 55 applies. For the parallel reduction, define $|t \cdot^{\text{par}} [\lambda x, y.q]| := |t|(2 + |q|)$.

◀

6.1 Strong Normalization of the detour conversion

We now prove strong normalization for $\longrightarrow_a$ by adapting the well-known *saturated sets method* of Tait [20] and Girard [8] to our calculus. Recall that Term is the set of all untyped proof-terms. (Definition 27.) We write SN for the set of strongly normalizing (untyped) terms and we write Var for the set of variables.

▶ **Definition 57.** **1.** The set Neut of *neutral terms* is defined by

a. Var $\subseteq$ Neut,

b. $t \cdot [\overline{p}\ ; \overline{\lambda y.q}] \in \mathsf{Neut}$ for all $t \in \mathsf{Neut}$ and $\overline{p}, \overline{\lambda y.q} \in \mathsf{SN}$.

2. The term t does a *key reduction* to t', notation $t \longrightarrow_a^k t'$, in case

a. t is a redex itself (according to Definition 29) and t' is its reduct,

b. $t = t_0 \cdot [\overline{p}\ ; \overline{\lambda y.q}]$, $t' = t_1 \cdot [\overline{p}\ ; \overline{\lambda y.q}]$ and $t_0 \longrightarrow_a^k t_1$.

3. A set $X \subseteq \mathsf{Term}$ is *saturated* ($X \in \mathsf{SAT}$) if it satisfies the following properties

a. $X \subseteq \mathsf{SN}$,

b. $\mathsf{Neut} \subseteq X$

c. X is closed under *key-redex expansion*: if $t \in \mathsf{SN}$ and $\forall q(t \longrightarrow_a^k q \Rightarrow q \in X)$, then $t \in X$.

4. For a connective c of arity n and $X_1, \ldots, X_n \in \mathsf{SAT}$ we define the set $c(X_1, \ldots, X_n)$ as follows. Assume that $r_1, \ldots, r_m$ are the elimination rules for c.

$$\begin{aligned} c(X_1, \ldots, X_n) := \{t \mid\ & \forall r_i \in \{r_1, \ldots, r_m\} \\ & \forall D \in \mathsf{SAT}, \forall \overline{p}, \overline{q} \in \mathsf{Term} \\ \forall k(p_k \in X_k) \wedge (\forall \ell \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)) \quad & \Longrightarrow \quad t \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}] \in D\ \} \end{aligned}$$

In the definition of $c(X_1, \ldots, X_n)$ it should be clear that we quantify over all elimination rules for the connective c. In the quantification $\forall \overline{p}, \overline{q} \in \mathsf{Term}$ we could also quantify over $\forall \overline{p}, \overline{q} \in \mathsf{SN}$: it amounts to the same because the additional conditions $\forall k(p_k \in X_k)$ and $\forall \ell \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D$ imply that $\overline{p}, \overline{q} \in \mathsf{SN}$.

▶ **Lemma 58.** *If* $X_1, \ldots, X_n \in \mathsf{SAT}$*, then* $c(X_1, \ldots, X_n) \in \mathsf{SAT}$.

Proof. We check the 3 conditions for $c(X_1, \ldots, X_n)$. Suppose $X_1, \ldots, X_n \in \mathsf{SAT}$.

a. That $c(X_1, \ldots, X_n) \subseteq \mathsf{SN}$ follows directly from the fact that if $t \in c(X_1, \ldots, X_n)$, then $t \cdot [\overline{p}\ ; \overline{\lambda x.q}] \in D$ and $D \subseteq \mathsf{SN}$, so $t \cdot [\overline{p}\ ; \overline{\lambda x.q}] \in \mathsf{SN}$, so $t \in \mathsf{SN}$.

b. For $t \in \mathsf{Neut}$ and $D \in \mathsf{SAT}$ and $\overline{p}, \overline{q} \in \mathsf{SN}$ with $\forall k(p_k \in X_k)$ and $\forall \ell \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$, we have $t \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}] \in \mathsf{Neut} \subseteq D$, so we can conclude that $t \in c(X_1, \ldots, X_n)$.

c. Suppose $t \in \mathsf{SN}$ and $\forall t'(t \longrightarrow_a^k t' \Rightarrow t' \in c(X_1, \ldots, X_n))$ (*). Let r_i be a rule for c and let $D \in \mathsf{SAT}$, $\overline{p}, \overline{q} \in \mathsf{Term}$ with $\forall k(p_k \in X_k)$ and $\forall \ell \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$. For all t' with $t \longrightarrow_a^k t'$ we have $t \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}] \longrightarrow_a^k t' \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}]$ and $t' \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}] \in D$ by (*). So, $t \cdot_{r_i} [\overline{p}\ ; \overline{\lambda y.q}] \in D$ and so $t \in c(X_1, \ldots, X_n)$.

◀

We use the saturated sets as a semantics for types: if A is a type, $\langle A \rangle$ will be a saturated set. The simplest way to do this is to interpret all type variables (proposition letters) as the set SN, which is indeed a saturated set.

▶ **Definition 59.** For A a type, we define $\langle A \rangle$ by induction on A as follows.

- $\langle A \rangle := \mathsf{SN}$ if A is a proposition letter.
- $c(A_1, \ldots, A_n) := c(\langle A_1 \rangle, \ldots, \langle A_n \rangle)$, where the right hand side is the interpretation of the connective c on saturated sets, as given in Definition 57, case (4).

We will often confuse A and $\langle A \rangle$, to avoid notational overhead, and just identify the proposition A with its interpretation as a saturated set $\langle A \rangle$.

▶ **Definition 60.** Given a context Γ, a map (valuation) $\rho : \mathsf{Var} \to \mathsf{Term}$ satisfies Γ, notation $\rho \models \Gamma$, in case $\rho(x) \in \langle A \rangle$ for all $x : A \in \Gamma$.

If $t \in \mathsf{Term}$ and $\rho : \mathsf{Var} \to \mathsf{Term}$, we write $\langle t \rangle_\rho$ for t where ρ has been carried out as a substitution on t.

A valuation $\rho : \mathsf{Var} \to \mathsf{Term}$ is only relevant for a finite number of variables: those that are declared in the context Γ under consideration. So we will always assume that $\rho(x) \neq x$ only for a finite number of $x \in \mathsf{Var}$. Those x we call the *support of* ρ. When applying ρ as a substitution to a term t we may need to "go under a λ", e.g. when applying ρ to $\{\overline{p} \,;\, \overline{\lambda x.q}\}$ In this case we always assume that the bound variable is not in the support of ρ. (We can always rename it.)

▶ **Lemma 61.** *If $\Gamma \vdash t : A$, and $\rho \models \Gamma$, then $\langle t\rangle_\rho \in \langle A\rangle$.*

Proof. By induction on the derivation of $\Gamma \vdash t : A$. Suppose $\rho \models \Gamma$. For the (axiom) case, it is trivial. We ignore ρ for the rest of the proof, as it gives a lot of notational overhead, so we just write t for $\langle t\rangle_\rho$.

- Suppose $\Phi = c(A_1, \ldots, A_n)$ and

$$\frac{\ldots \Gamma \vdash s_j : A_j \ldots \qquad \ldots \Gamma, x_i : A_i \vdash t_i : \Phi \ldots}{\Gamma \vdash \{\overline{s} \,;\, \overline{\lambda x.t}\}_r : \Phi}\ \text{in}$$

  Let r' be a rule for c, $D \in \mathsf{SAT}$, $\overline{p}, \overline{q} \in \mathsf{Term}$ with $\forall k(p_k \in A_k)$ and $\forall \ell \forall u_\ell \in A_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$. For $\{\overline{s} \,;\, \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}]$ there are the following possible key-reductions:

$$\{\overline{s} \,;\, \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}] \quad \longrightarrow_a^k \quad q_l[y_l := s_j] \tag{1}$$

$$\{\overline{s} \,;\, \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}] \quad \longrightarrow_a^k \quad t_i[x_i := p_k] \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}] \tag{2}$$

  In case (1), $q_l[y_l := s_j] \in D$ by the assumption and the induction hypothesis. In case (2), $t_i[x_i := p_k] \in \Phi$ by the induction hypothesis and so $t_i[x_i := p_k] \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}] \in D$ by the definition of $\Phi = c(A_1, \ldots, A_n)$ as a saturated set. So, $\{\overline{s} \,;\, \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p} \,;\, \overline{\lambda y.q}] \in \mathsf{SN}$ and all its key reductions are in D, so the term is in D. Therefore, $\{\overline{s} \,;\, \overline{\lambda x.t}\}_r \in \Phi$.
- Suppose $\Phi = c(A_1, \ldots, A_n)$ and

$$\frac{\Gamma \vdash t : \Phi \qquad \ldots \Gamma \vdash p_k : A_k \ldots \qquad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash t \cdot_r [\overline{p} \,;\, \overline{\lambda y.q}] : D}\ \text{el}$$

  Then $t \cdot_r [\overline{p} \,;\, \overline{\lambda y.q}] = t \cdot_r [\overline{p} \,;\, \overline{\lambda y.q}] \in D$ by $t \in \Phi = c(A_1, \ldots, A_n)$ and the definition of $c(A_1, \ldots, A_n)$ as a saturated set and the induction hypothesis.

◀

The following is now an immediate corollary by taking $\rho(x) := x$ for all $x \in \mathsf{Var}$. Because $\mathsf{Var} \subseteq \mathsf{Neut} \subseteq \langle A\rangle$, we know that $\rho \models \Gamma$. So, if $\Gamma \vdash t : A$, then $\langle t\rangle_\rho = t \in \langle A\rangle \subseteq \mathsf{SN}$.

▶ **Theorem 62.** *The reduction $\longrightarrow_a$ is strongly normalizing: all $\longrightarrow_a$-reductions on proof terms are finite.*

▶ **Corollary 63.** *The reduction $\longrightarrow_a$ for the optimized rules of Definition 36, the parallel $\wedge$-elimination rule of Definition 44, the standard $\to$-introduction of Definition 47 and the traditional rule for $\neg$-elimination of Definition 50 are strongly normalizing.*

Proof. By Theorem 62 and the fact that reduction is preserved by the translation: Lemmas 43, 46 and 49. ◀

6.2 Weak Normalization of conversion

We now give a strategy for finding a normal form for the combined $\longrightarrow_{ab}$ reduction, the union of $\longrightarrow_a$ and $\longrightarrow_b$. This proves that $\longrightarrow_{ab}$ is weakly normalizing and it also gives a concrete procedure for finding a normal form. Due to the fact that, in general, reduction is not confluent, this normal form is not unique, but it does yield *decidability* via the *sub-formula property*. The weak normalization proof follows the well-known idea, originally due to Turing (see [5]) for simple type theory, to *contract the innermost redex of highest rank*.

▶ **Definition 64.** We define the *rank of a formula* A, $\mathsf{rk}(A)$ as follows.

- $\mathsf{rk}(A) := 1$ if A is a proposition letter.
- $\mathsf{rk}(c(A_1, \ldots, A_n) := 1 + \max\{\mathsf{rk}(A_1), \ldots, \mathsf{rk}(A_n)\}$ if c is a connective of arity n.

We define the *rank of a redex* as follows.

- The rank of $\{\overline{p} \,;\, \overline{\lambda x.q}\}_{r'} \cdot_r [\overline{s} \,;\, \overline{\lambda y.r}]$ is the rank of the type of $\{\overline{p} \,;\, \overline{\lambda x.q}\}_{r'}$.
- The rank of $(t \cdot_{r'} [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot_r [\overline{s} \,;\, \overline{\lambda y.r}]$ is the rank of the type of $t \cdot_{r'} [\overline{p} \,;\, \overline{\lambda x.q}]$.

We will sometimes mark the redex with its type Φ such that $\mathsf{rk}(\Phi)$ is the rank of the redex. We do this by writing Φ as a superscript to the elimination constructor. To clarify, we summarize again the possible reduction steps of the form $\longrightarrow_a$ and $\longrightarrow_b$.

▶ Notation 65. From Definition 29, we have the reduction $\longrightarrow_a$ and form Definition 32 we have the reduction $\longrightarrow_b$. We introduce the following notation.

$$
\begin{aligned}
\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] &\quad \longrightarrow_{a1} \quad r_\ell[y_\ell := p_j] \\
\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] &\quad \longrightarrow_{a2} \quad q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \\
(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] &\quad \longrightarrow_{b} \quad t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]
\end{aligned}
$$

Here, the proviso's of Definition 29 apply, so the first is the "$\ell = j$ case" which we will call $\longrightarrow_{a1}$, and the second is the "$k = i$ case" which we will call $\longrightarrow_{a2}$.

We give two Lemmas that show that the creation of new redexes is limited.

▶ **Lemma 66.** **1.** *If $t \longrightarrow_b t'$ by contracting a redex of $\mathsf{rk}(\Phi)$ then the newly created redexes are also of $\mathsf{rk}(\Phi)$.*

2. *Suppose $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$. If $q_i[x_i := s_k]$ is an introduction term (that is: $q_i[x_i := s_k]$ is of the form $\{\ldots \,;\, \ldots\}$), then q_i is an introduction term. Similarly, if $q_i[x_i := s_k]$ is an elimination term (that is: $q_i[x_i := s_k]$ is of the form $\ldots \cdot [\ldots \,;\, \ldots]$), then q_i is an elimination term.*

Proof. **1.** If $t \longrightarrow_b t'$ by contracting a redex of $\mathsf{rk}(\Phi)$, then t contains a sub-term $s \cdot [\overline{p} \,;\, \overline{\lambda x.q}] \cdot^{\Phi} [\overline{u} \,;\, \overline{\lambda y.r}]$ which is contracted to $s \cdot [\overline{p} \,;\, \overline{\lambda x.q \cdot^{\Phi} [\overline{u} \,;\, \overline{\lambda y.r}]}]$. The newly created redexes (if any) are all of $\mathsf{rk}(\Phi)$.

2. Suppose $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$. Then $q_i : \Phi$ and $s_k : A_k$ which is a sub-formula of Φ, as $\Phi = c(A_1, \ldots, A_n)$. If $q_i[x_i := s_k]$ is an introduction term, then either q_i is an introduction term itself or $q_i = x_i$ and s_k is an introduction term. The latter case can only occur if $s_k : \Phi$, but it is not, because its type is a sub-formula of Φ. So q_i is an introduction term. The case for $q_i[x_i := s_k]$ being an elimination term is similar.

◀

The Lemma states that both the newly created redexes due to $\longrightarrow_b$ and $\longrightarrow_{a2}$ are already "hidden" inside the term. We give a list of facts about redex creation and the ranks of redexes.

▶ Fact 67. **1.** A reduction step can produce more redexes either by (i) *copying existing redexes* or by (ii) *creating new redexes*. Copying occurs through substitution, in a reduction step $\longrightarrow_{a1}$ or $\longrightarrow_{a2}$.

2. Creating new redexes happens in either one of the following ways.

a. When doing an $\longrightarrow_a$ step: in a sub-term $x \cdot [\overline{p} \,;\, \overline{\lambda y.q}]$, we substitute $\{\overline{s} \,;\, \overline{\lambda z.r}\}$ for x. This creates an a-redex of lower rank.

b. When doing an $\longrightarrow_a$ step: in a sub-term $x \cdot [\overline{p} \,;\, \overline{\lambda y.q}]$, we substitute $t \cdot [\overline{s} \,;\, \overline{\lambda z.r}]$ for x This creates a b-redex of lower rank.

c. When $\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_{a1} r_\ell[y_\ell := p_j]$ where this term occurs as a sub-term: $r_\ell[y_\ell := p_j] \cdot^{\Psi} [\ldots \,;\, \ldots]$ and $r_\ell[y_\ell := p_j] = \{\ldots \,;\, \ldots\}$.
This creates a new a-redex of unrelated rank.

d. When $\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_{a1} r_\ell[y_\ell := p_j]$ where this term occurs as a sub-term: $r_\ell[y_\ell := p_j] \cdot^{\Psi} [\ldots \,;\, \ldots]$ and $r_\ell[y_\ell := p_j] = \ldots \cdot [\ldots \,;\, \ldots]$.
This creates a new b-redex of unrelated rank.

e. When $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$, where $q_i = \{\ldots \,;\, \ldots\}$.
This creates a new a-redex of the same rank.

f. When $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$, where $q_i = \ldots \cdot [\ldots \,;\, \ldots]$.
This creates a new b-redex of the same rank.

g. If $(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] \longrightarrow_b t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]$, where $q_i = \{\ldots \,;\, \ldots\}$.
This creates a new a-redex (possibly more) of the same rank.

h. If $(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] \longrightarrow_b t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]$, where $q_i = \ldots \cdot [\ldots \,;\, \ldots]$.
This creates a new b-redex (possibly more) of the same rank.

Note that in the cases **e** and **f** of Fact 67 we use the second part of Lemma 66.

The idea is to contract an innermost redex of highest rank of a term in b-normal form (that is: a term that cannot do a $\longrightarrow_b$-step). The advantage of b-normal forms is that cases **c** and **d** of the Fact 67 do not occur. (Because in these cases, the term one starts with is not in b-normal form.)

▶ **Lemma 68.** *If f is a well-typed term in b-normal form that has one redex of maximum rank, say R, then f can be reduced to a term f' in b-normal form that has maximum rank below R.*

Proof. By induction on the size of f.

1. If $f = \{\overline{p} \,;\, \overline{\lambda x.q}\}$ or $f = x \cdot [\overline{p} \,;\, \overline{\lambda x.q}]$ or $f = \{\overline{p} \,;\, \overline{\lambda x.q}\} \cdot [\overline{s} \,;\, \overline{\lambda y.r}]$ and the redex of highest rank is inside $\overline{p}$, $\overline{q}$, $\overline{s}$ or $\overline{r}$, then we are done by the induction hypothesis.

2. Suppose $f = \{\overline{p} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}]$ is itself a redex of highest rank, $\mathsf{rk}(\Phi)$. We look at the possible ways in which a new redex may arise, following Fact 67. The cases **c**, **d**, **g** and **h** don't apply.

- For case **a**: the newly created redexes are of lower rank and the resulting term is in b-nf.
- For case **b**: the newly created redexes are of lower rank. The resulting term may not be in b-nf, but we can contract all the newly created b-redexes to obtain a b-normal form. According to Lemma 66, case (1), this does not create new redexes of higher rank, so we are done.
- For case **e**: $f = \{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$ with $q_i = \{\ldots \,;\, \ldots\}$. By induction hypothesis, $q_i \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \twoheadrightarrow g$ for some g in b-normal form with all redexes of lower rank. (Note that $q_i \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$ is in b-normal form.) Then $q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \twoheadrightarrow g[x_i := s_k]$ and due to the fact that the type of s_k is a sub-formula of Φ, this only contains new redexes of lower rank, so we are done.

- For case **f**: $f = \{\overline{p}\ ;\ \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}]$ with $q_i = t \cdot [\overline{u}\ ;\ \overline{\lambda z.v}]$. If we take g to be the b-normal form of $q_i \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}]$, this term contains disjoint sub-terms of the shape $\lambda w.d \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}]$ that all have one maximal redex of rank R and that have length smaller than the length of f. By induction hypothesis, these can all be reduced to terms with only redexes of lower rank. Having done this, we obtain g as a reduct of $q_i \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}]$ that is in b-normal form and contains only redexes of rank lower than R. To obtain f', we notice that $f \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k}\ ;\ \overline{\lambda y.r}] \twoheadrightarrow g'[x_i := s_k]$, which only contains b-redexs of lower rank, so we can take f' to be the b-normal form of $g'[x_i := s_k]$.

◀

▶ **Theorem 69.** *For any set of connectives $\mathcal{C}$, the reduction $\longrightarrow_{ab}$ of the calculus $\lambda^{\mathcal{C}}$ is weakly normalizing and we have a procedure to compute a normal form for a well-typed term.*

Proof. We consider the following measure $\mathsf{m}(-)$ terms: $\mathsf{m}(t) := (R, m)$, where R is the maximal rank of a redex in t and m is the number of redexes of rank R in t. We consider this measure under the lexicographic ordering.

Given a term t, we first compute its b-normal form, t_1 and consider $\mathsf{m}(t_1) = (R, m)$. Then we pick p, an innermost redex of maximal rank inside t_1. Following Lemma 68, we reduce p to p', in which all redexes are of rank below R. We do this reduction on t_1, obtaining t_2. (So $t_1 \twoheadrightarrow t_2$.) Notice that $\mathsf{m}(t_1) > \mathsf{m}(t_2)$. We continue in this way, obtaining a normal form of t, because the lexicographic ordering is well-founded. ◀

We recall Lemma 35 which describes NF inductively, the set of terms in normal form. If t is in normal form, then t is of either one of the following three forms

1. t is a variable,
2. $t = \{\overline{p}\ ;\ \overline{\lambda y.q}\}$, with all p_i and q_j in normal form,
3. $t = x \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]$, with x a variable and all p_i and q_j in normal form.

6.3 Corollaries of normalization

▶ **Theorem 70.** *For any set of connectives $\mathcal{C}$, the calculus $\lambda^{\mathcal{C}}$ is consistent, that is: there are types A for which there is no closed term t with $\vdash t : A$.*

Proof. Take A to be a propositional variable and suppose $\vdash t : A$ with t in normal form. The three possible cases for t are given in Lemma 35, which we have recalled above. The first and third case are impossible, because t cannot contain any free variable. The second case is impossible, because an introduction term is always of a composite type. ◀

The calculus (and logic) $\lambda^{\mathcal{C}}$ also satisfies the sub-formula property.

▶ **Theorem 71.** *Given a set of connectives $\mathcal{C}$, the calculus $\lambda^{\mathcal{C}}$ satisfies the* sub-formula property*, that is: if $\Gamma \vdash t : A$, then there is a term t' such that $\Gamma \vdash t' : A$ and all types of all sub-terms of t' are either sub-types of A or of some A_i for a declaration $x_i : A_i$ in Γ.*

Proof. If $\Gamma \vdash t : A$, then (by Theorem 69) there is a term t' in normal form with $\Gamma \vdash t' : A$. We use Lemma 35 and prove by induction on t' that "all types of all sub-terms of t' are either sub-types of A or of some A_i for a declaration $x_i : A_i$ in Γ". For simplicity we abbreviate this property to "t' *satisfies the sub-type property for* $\Gamma; A$".

- $t' = x$, a variable. Then we are done.

- $t' = \{\overline{p}\ ;\ \overline{\lambda x.q}\}$, an introduction term. Then by induction hypothesis, all sub-terms of $\overline{p}$ satisfy the sub-type property for $\Gamma; A_i$ for some A_i which is a sub-type of A. For the $\lambda x_j.q_j$ in $\overline{\lambda x.q}$, we have $\Gamma, x_j : A_j \vdash q_j : A$ for some A_j which is a sub-type of A. By induction hypothesis, for all j, all sub-terms of q_j satisfy the sub-type property for $\Gamma, x_j : A_j; A$. So all sub-terms of $\overline{\lambda x.q}$ satisfy the sub-type property for $\Gamma; A$ and we are done.
- $t' = x \cdot [\overline{p}\ ;\ \overline{\lambda x.q}]$, an elimination term. Suppose $x : C$. Each p_i is of type B_i for some sub-type B_i of C, so the induction hypothesis yields that all sub-terms of $\overline{p}$ satisfy the sub-type property for $\Gamma; A$. For the $\lambda x_j.q_j$ in $\overline{\lambda x.q}$, we have $\Gamma, x_j : B_j \vdash q_j : A$ for some B_j which is a sub-type of C. By induction hypothesis, for all j, all sub-terms of q_j satisfy the sub-type property for $\Gamma, x_j : B_j; A$. So all sub-terms of $\overline{\lambda x.q}$ satisfy the sub-type property for $\Gamma; A$ and we are done.

◀

▶ **Theorem 72.** *In $\lambda^{\mathcal{C}}$, given a context Γ and a type D, the problem $\Gamma \vdash ? : D$ is decidable. That is, it is whether there is a term t for which $\Gamma \vdash t : D$.*

Proof. By Theorem 69 we can limit our search to a term in normal form. So we can restrict the elimination rules to the following restricted case, where $\Phi = c(A_1, \ldots, A_n)$. (Compare with the original rules in Definition 27.)

$$\frac{x : \Phi \in \Gamma \quad \ldots \Gamma \vdash p_k : A_k \ldots \quad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash x \cdot_r [\overline{p}\ ;\ \overline{\lambda y.q}] : D}\ \text{el}$$

Now, given Γ and D, the following algorithm searches a term t in normal form with $\Gamma \vdash t : D$. (1) Check if $x : D \in \Gamma$ for some x and otherwise (2) try an introduction rule (in case D is composite) and (3) try an elimination rule for each $x : \Phi \in \Gamma$ with Φ a composite formula. In the recursive case, this gives finitely many possibilities to try and each try creates new goals of the form $\Gamma, y_j : A_j \vdash ? : D$ or of the form $\Gamma \vdash ? : A_i$ with A_j and A_i sub-formulas of Γ, D. This search terminates because the number of sub-formulas in the context increases (which is bound by the number of all sub-formulas of Γ, D), and otherwise the size of the goal-formula decreases.

◀

As a corollary, we find that all the variants of the logical rules we have considered are decidable and consistent, simply because they are (with respect to derivability) equivalent to the set of rules for $\wedge, \vee, \rightarrow, \neg, \bot, \top$ that we extract from the truth tables, for which Theorems 70 and 72 apply. We can also say a bit more about the conversion of derivations in these systems themselves: detour conversion is strongly normalizing, permutation conversion is strongly normalizing and we can also conclude weak normalization of the combined conversion.

▶ **Theorem 73.** *The reductions for the optimized rules of Definition 36, the parallel $\wedge$-elimination rule of Definition 44, the standard $\rightarrow$-introduction of Definition 47 and the traditional rule for $\neg$-elimination of Definition 50 are weakly normalizing.*

Proof. The proof follows the same argument as the proof of Theorem 69. The crucial Lemmas are Lemmas 68 and 66, which can be proved again with the reduction rules mentioned in the statement of Theorem 73 added. Furthermore, the permutation conversion, $\longrightarrow_b$ is strongly normalizing. (Corollary 56.) ◀

7 Conclusion and Further work

We have studied the general procedure for deriving intuitionistic natural deduction rules from truth tables, that we have presented in [7]. We have defined detour conversion and permutation in general and we have proven that both are strongly normalizing and that the combination of the two is weakly normalizing. We have done so by defining a proof-term calculus for derivations on which we have defined the reduction rules that correspond to conversion of derivations. This follows the well-known Curry-Howard formulas-as-types isomorphism that establishes an isomorphism between proofs (derivations in natural deduction) and terms. We have shown that very many well-known formalisms for intuitionistic natural deduction can be defined in terms of our calculus, including the conversion rules for derivations. Our paper also provides a straightforward method for deriving a term calculus for any connective that is given via a truth table: the term constructions and reduction rules are self-contained and normalizing by construction. We have shown this on various examples, most notably the nand-connective.

The work described here leaves various questions unanswered. For example, is proof normalization (the combination of detour conversion and permutation conversion) strongly normalizing in general for an arbitrary set of connectives? We would believe so, but have not yet proved it. Techniques as in [9], where this property is proved for intuitionistic logic, may be useful.

It also raises various new research questions: The rules are not Church-Rosser (confluent) in general, but one may wonder whether there is a certain condition that guarantees confluence. We have seen in Examples 23, 30 and 40 that fixing a choice for the "matching case" in a detour convertibility may render the reduction confluent. It is not clear if this would work in general.

Another topic to look into is detour conversion for the classical case, and what its connection is with known term calculi for classical logic, for example as studied in [13], [1] and [2]. Also, it might be interesting to look at these general rules from a linear perspective: what if we enforce the rules to be linear?

Finally, we may wonder whether our research could contribute to the study of "harmony in logic", as first introduced by Prawitz [15] and further studied by various authors like [16, 12, 23, 4, 3]. The inversion principle explains the elimination rules as capturing the "least information" that is conveyed by the introduction rules. This can also be dualized (as is done in [12] in their "uniform calculus") by explaining the introduction rules in terms of the elimination rules. It would be interesting to study the relation with our rules, where there is no a priori preference for the introduction or elimination rules.

From our research, we would propose the following as a proper system for intuitionistic logic with "parallel elimination rules" that follow Prawitz' [15] inversion principle. These rules are derived from the truth tables and optimized following Lemma 9, but not using Lemma 12. Compare with Definition 13; the special rules are $\wedge$-elimination and $\to$-elimination.

▶ **Definition 74.** The *parallel elimination rules* for the intuitionistic propositional connectives

$\wedge, \vee, \to, \neg, \bot$ and $\top$ are given below.

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\,\wedge\text{-in} \qquad \frac{\vdash A \wedge B \quad A \vdash D}{\vdash D}\,\wedge\text{-el}_{0\_} \qquad \frac{\vdash A \wedge B \quad B \vdash D}{\vdash D}\,\wedge\text{-el\_}_{0}$$

$$\frac{\vdash A}{\vdash A \vee B}\,\vee\text{-inl} \qquad \frac{\vdash B}{\vdash A \vee B}\,\vee\text{-inr} \qquad \frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\,\vee\text{-el}$$

$$\frac{A \vdash A \to B}{\vdash A \to B}\,\to\text{-in}_a \qquad \frac{\vdash B}{\vdash A \to B}\,\to\text{-in}_b \qquad \frac{\vdash A \to B \quad \vdash A \quad B \vdash D}{\vdash D}\,\to\text{-el}$$

$$\frac{A \vdash \neg A}{\vdash \neg A}\,\neg\text{-in} \qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash D}\,\neg\text{-el} \qquad \frac{}{\vdash \top}\,\top\text{-in} \qquad \frac{\vdash \bot}{\vdash D}\,\bot\text{-el}$$

8 References

References

1. Z. Ariola and H. Herbelin. Minimal classical logic and control operators. In *ICALP*, volume 2719 of *LNCS*, pages 871–885. Springer, 2003.
2. P.-L. Curien and H. Herbelin. The duality of computation. In *ICFP*, pages 233–243, 2000.
3. R. Dyckhoff. Some remarks on proof-theoretic semantics. In *Advances in Proof-Theoretic Semantics*, pages 79–93. Springer, 2016.
4. N. Francez and R. Dyckhoff. A note on harmony. *Journal of Philosophical Logic*, 41(3):613–628, 2012. URL: `http://dx.doi.org/10.1007/s10992-011-9208-0`, `doi:10.1007/s10992-011-9208-0`.
5. R.O. Gandy. An early proof of normalization by A.M. Turing. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, page 453–455. Academic Press Limited, 1980.
6. G. Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, pages 176–210, 405–431, 1935. English translation in [19].
7. H. Geuvers and T. Hurkens. Deriving natural deduction rules from truth tables. In *ICLA*, volume 10119 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2017.
8. J.-Y. Girard et al. *Proofs and types*, volume 7 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, Cambridge, 1989.
9. F. Joachimski and R. Matthes. Short proofs of normalization for the simply- typed lambda-calculus, permutative conversions and Gödel's T. *Arch. Math. Log.*, 42(1):59–87, 2003.
10. E.G.K López-Escobar. Standardizing the N systems of Gentzen. In *Models, Algebras and Proofs*, volume 203 of *Lecture Notes in Pure and Applied Mathematics*, page 411–434. Marcel Dekker Inc., New York, 1999.
11. P. Milne. Inversion principles and introduction rules. In *Dag Prawitz on Proofs and Meaning*, volume 7 of *Outstanding Contributions to Logic*, pages 189–224. Springer, 2015.
12. S. Negri and J. von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.
13. M. Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.
14. D. Prawitz. *Natural deduction: a proof-theoretical study*. Almqvist & Wiksell, 1965.
15. D. Prawitz. Ideas and results in proof theory. In J. Fenstad, editor, *2nd Scandinavian Logic Symposium*, pages 237–309. North-Holland, 1971.

16 S. Read. Harmony and autonomy in classical logic. *J. Philosophical Logic*, 29(2):123–154, 2000. URL: `https://doi.org/10.1023/A:1004787622057`, `doi:10.1023/A:1004787622057`.

17 P. Schroeder-Heister. A natural extension of natural deduction. *J. Symb. Log.*, 49(4):1284–1300, 1984.

18 Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science Inc., New York, NY, USA, 2006.

19 M.E. Szabo. *The Collected Papers of Gerhard Gentzen*. North-Holland, Amsterdam, 1969.

20 W.W. Tait. Intensional interpretations of functionals of finite type I. *J. Symb. Log.*, 32(2):198–212, 1967. URL: `http://dx.doi.org/10.2307/2271658`, `doi:10.2307/2271658`.

21 N. Tennant. Ultimate normal forms for parallelized natural deductions. *Logic Journal of the IGPL*, 10(3):299–337, 2002.

22 D. van Dalen. *Logic and structure (3. ed.)*. Universitext. Springer, 1994.

23 J. von Plato. Natural deduction with general elimination rules. *Arch. Math. Log.*, 40(7):541–567, 2001.